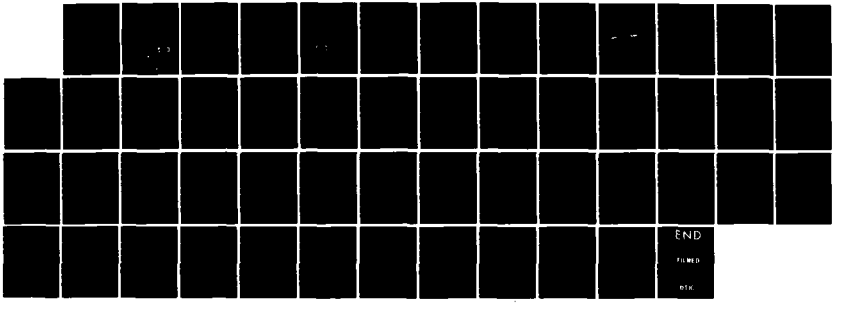
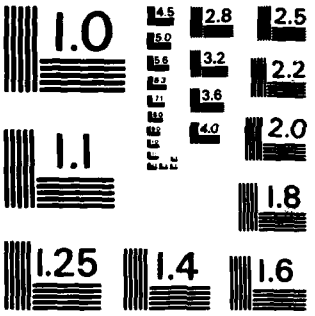
AD-A147 887 PLOT3D: AN INTERACTIVE GRAPHICS CODE FOR THREE DIMENSIONAL PLOTS(U) NAVAL RESEARCH LAB WASHINGTON DC G A KERAMIDAS ET AL. 28 SEP 84 NRL-MR-5410 1/1

UNCLASSIFIED F/G 9/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

NRL Memorandum Report 5410

PLOT3D
An Interactive Graphics Code For Three Dimensional Plots

G. A. KERAMIDAS AND E. W. MINER

Fluid Dynamics Branch
Marine Technology Division

W. BAUMAN

Naval Seas Systems Command

September 28, 1984

NAVAL RESEARCH LABORATORY
Washington, D.C.

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S) NRL Memorandum Report 5410	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Research Laboratory 6b OFFICE SYMBOL (If applicable) Code 5841	7a NAME OF MONITORING ORGANIZATION
6c ADDRESS (City, State and ZIP Code) Washington, DC 20375-5000	7b ADDRESS (City, State and ZIP Code)
8a NAME OF FUNDING/SPONSORING ORGANIZATION Office of Naval Research 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State and ZIP Code) Arlington, VA	10 SOURCE OF FUNDING NOS PROGRAM ELEMENT NO: 61153N PROJECT NO: TASK NO: RR023-01-41 WORK UNIT NO: 58-1470-04
11 TITLE (Include Security Classification) (see Page ii)	
12 PERSONAL AUTHOR(S) Keramidas, G.A.; Miner, E.W.; and Bauman, W.	
13a TYPE OF REPORT Final 13b TIME COVERED FROM TO	14 DATE OF REPORT (Yr., Mo., Day) 1984 September 28 15 PAGE COUNT 52
16 SUPPLEMENTARY NOTATION	
17 COSATI CODES FIELD / GROUP / SUB GR	18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Graphics Hidden line 3-dimensional

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This report decribes the basic operation procedure, structure and use of the PLOT3D computer graphics code. A description of the code is given in the first part of this manual followed by the description of the input parameters and the requirements for using the code. Some samples of the capabilities of the code are given at the end of this report.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS ☐	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL G. A. Keramidas	22b TELEPHONE NUMBER (Include Area Code) 22c OFFICE SYMBOL Code 5841

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE

11. TITLE (include Security Classification)

PLOT3D An Interactive Graphics Code for Three Dimensional Plots

CONTENTS

PLOT3D
AN INTERACTIVE GRAPHICS CODE FOR THREE DIMENSIONAL PLOTS

INTRODUCTION

The PLOT3D graphics computer code provides a user with an interactive and powerful tool for quickly analyzing three dimensional functional data. The code allows the user to choose from a variety of options or parameters in selecting an appropriate display of data. The flexibility of the code should be sufficient to satisfy a wide range of user needs. The code runs in an interactive mode so that the user can easily obtain the desired display of data. All options and parameters (such as hidden line removal) are selected from one of the two menus in the program.

The code plots the user's data as if the data were points on a three-dimensional surface or solid. The code then permits the user to change the orientation from which the data are viewed, the scaling of the data, the frequency of the points that are plotted and other parameters in the display of the data. The code is designed so that a user will first develop the display on a graphics terminal screen and then plot the data on a pen plotter.

The code is written in FORTRAN 77 and can be used on any HP 1000 computer which has an HP graphics terminal and the GRAPHICS 1000-II DGL software.

This computer code was developed as part of the hydrodynamics program of the Fluid Dynamics Branch and has been a useful tool for the graphical presentation of results.

1. CODE DESCRIPTION

The development of the PLOT3D graphics computer code has been based on two existing subroutines; which perform the coordinate transformation and plotting of data. Certain modifications were necessary to these routines in order to conform with program and system requirements. One of these subroutines performs the actual plotting of data and hidden line removal and was originally developed by Watkins in a 3-D plotting package.[1]

The PLOT3D code was implemented on the HP 1000 computer system and the code is operational under various operating systems.

The PLOT3D code is composed of three modules. The main program is named PLT3D and controls the use of the two segments PLT3T and PLT3P. The segment PLT3T plots to the terminal screen and provides all the interactive options. The segment PLT3P is used to plot the last screen plot on the pen plotter. These modules are described below.

1.1. PLT3D

This is the main program of the computer code and is designed as the driver for the two graphics modules. At the end of a graphic task, control is transferred to this driver which determines the next

Manuscript approved June 18, 1984.

task required. The transfer between main program and the other two modules is achieved by standard main-to-segment or segment-to-segment control calls. All input and program parameters retain their values as they were last defined. This allows for a transfer of a graphics image from a graphics terminal screen to a pen plotter without any additional information.

1.2. PLT3T

This segment of the code is the largest in size since it contains most of the routines for program input-output control plus the routines for performing the graphics output on a graphics terminal. The names and description of the routines contained in this segment follow

PLT3T	MAIN routine of this program segment. Provides parameter definition and input of data for plotting
DMENU	Clears the screen and gives the menu for changing program parameters
DFUNCT	Menu with options used by DMENU
PMENU	Clears the screen and gives the menu for program control
PFUNCT	Option menu used by PMENU
ADISP	Clears alphanumeric display
GDISP	Clears graphics display
PLPAR	Contains the interactive input for all program parameters
PLOTF	Plots the function $y=f(x,z)$ according to the specified programs parameters.
PLOTS	Manipulates data for scaling and angles of rotation; Calls PLOTF to plot each line
PERIM	Draws the plane of projection and perimeter around the plotted data
PLTIF	Puts data information on the same figure as the plot
DATEP	Puts the date on the plot
REFAX	Draws the orientation of the axes on the plot
BOXIN	Draws a box around the figure
TITLE	Puts a figure title in lower part of page
ERRORS	Error output routine
VIEWP	Defines window and viewport for graphics output. Also defines aspect ratio and character size
DEVON	Initializes device and software for graphics display
VPMAX	Changes current viewport to the maximum for alphanumeric output
DEVOF	Turns off graphics device
LETTR	Text output routine
NUMBR	Numerical text output

1.3. PLT3P

This is the third segment of the code and it contains only routines necessary to perform graphics output on a hard copy device (pen plotter).

NAME	Description
PLT3P	Main program same function as in PLT3T
ADISP	Clears alphanumeric display
PLOTS	Same as in PLT3T segment
PLOTF	Plots on pen plotter instead of the graphics terminal
PERIM	Same as in PLT3T segment
PLTIF	Same as in PLT3T
DATEP	Same as in PLT3T
REFAX	Same as in PLT3T

BOXIN	Same as in PLT3T
TITLE	Same as in PLT3T
DEVON	Initializes graphics device (pen ploter) and selects pen color and speed
ERRORS	Same as in PLT3T
VIEW	Same as in PLT3T
VPMAX	Same as in PLT3T
DEVOF	Same as in PLT3T
LETTR	Same as in PLT3T
NUMBR	Same as in PLT3T

2. CODE OPERATION

Depending upon how the PLOT3D code has been installed on the system (see Section3) the user will access the program by the command

RU, PLT3D

The program displays a banner to the terminal and then calls segment PLT3T. PLT3T first requests a color value as shown in Fig. 1. A value between 1 and 5 is appropriate and will set the plotting color on the HP-2627 terminals. On other terminals (eg. HP-2648) the color parameter changes the plotting line style and a value of 1 is most appropriate.

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXX                                            XXXXX
XXXXX      * * * * *   PLOT 3-D   * * * * *      XXXXX
XXXXX                                            XXXXX
XXXXX This program will plot a three dimensional  XXXXX
XXXXX surface on the Screen or on the pen Plotter. XXXXX
XXXXX                                            XXXXX
XXXXX The default values of the plotting parametersXXXXX
XXXXX have been set for a fast plot on the screen  XXXXX
XXXXX After this fast plot the user can change the XXXXX
XXXXX parameters for the plot desired             XXXXX
XXXXX                                            XXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
ENTER VALUE FOR COLOR 1
```

Figure 1

PLT3T then prompts for the name of the file that contains the data to be plotted. In the current program the file name must be six characters or less. PLT3T then prompts for the number of data points in the file in the *x* and *z* directions. These prompts are shown in Fig. 2. The program reads the data to be plotted in the following way:

```
      READ (5)    (XPL(I), I = 1, NXPL)
      READ (5)    (ZPL(J), J = 1, NZPL)
      DO 100      I = 1, NXPL
      READ (5)    (YPL(I,J), J = 1, NZPL)
100   CONTINUE
```

and the data must have been written to the file exactly the same way and the maximum values for NXPL and NZPL are currently set to 151. The value of NXPL does not have to be the same as the

value of NZPL. Furthermore the XPL (I) or ZPL (I) do not have to be uniformly spaced. PLT3T then plots the data at the user's terminal using the program's default parameters. Before the plotting of data the program will prompt for input of the axis labels 24 characters each and for two lines of figure caption 52 characters each. A sample plot is shown in Fig. 3. After each plot to the terminal, the program pauses (with a request that "/" be entered before displaying the selection menus).

```
**************************************************
**** SPECIFY THE NAME OF THE DATA FILE FROM WHICH   ****
**** THIS PROGRAM WILL READ THE DATA TO BE PLOTTED  ****
**** FILE NAME IFILE =ELL05
**************************************************
**** SPECIFY THE NUMBER OF DATA POINTS FOR THE      ****
**** X(NXPL) ARRAY.  NXPL =101
**************************************************
**** SPECIFY THE NUMBER OF DATA POINTS FOR THE      ****
**** Z(NZPL) ARRAY.  NZPL =101
```

Fig. 2 — File name input and number of points

When the user is ready to continue, the program displays the data menu, with eleven options, as shown in Fig. 4. The parameters that PLT3T uses in plotting the data may be changed by selecting options 1-9. Option 10 causes the display of the program menu and option 11 causes control to return to the main program. Options 1-11 will be discussed in turn. For options 1-9, typing in a "/" in response to the prompt will preserve the existing value of the parameter. After a value is entered for a parameter, the user may continue changing more parameter values until option 10 or 11 is choosen.

Selecting option 1 clears the screen and prompts for entry of plotting frequency parameters is the x and z directions. These prompts are shown in Fig. 5. The default values of IPLX and IPLZ are 2 and every other point is plotted. Every tenth point would be plotted if IPLX or IPLZ were given a value of 10.

Option 2 of the data menu allows the user to reset the values of NXPL and NZPL — see Fig. 6. The number of points in either direction that are plotted can be reduced starting always from point one — but not increased, from numbers specified when the file is read in.

Option 3 prompts for new values of DELTA, THETA and PHI, as shown in Fig. 7. DELTA controls the vertical location of the plot on the screen and normally a value of 0.0 is appropriate. THETA is the angle (in degrees) that the figure is rotated (counter-clockwise) around the x-axis. Note the reference axis configuration shown in Fig. 3. A positive value of THETA rotates the y-axis toward the z-axis around the x-axis. Similarly a positive value of PHI rotates the z-axis toward the x-axis around the y-axis.

Figure 8 shows the prompts for changing the data scaling parameters (Option 4). The default values of 7, 2 and 5 will usually fill the plotting area. Increasing the values will enlarge the plot and decreasing the values will decrease the size of the plot. If the size of the plot is too large — which can occur with some scale factors and orientations — the figure would exceed the allowable plotting area and the program omits the plot. In such a case, the scale factors should be reduced so that the figure will fit in the allowable plotting area.

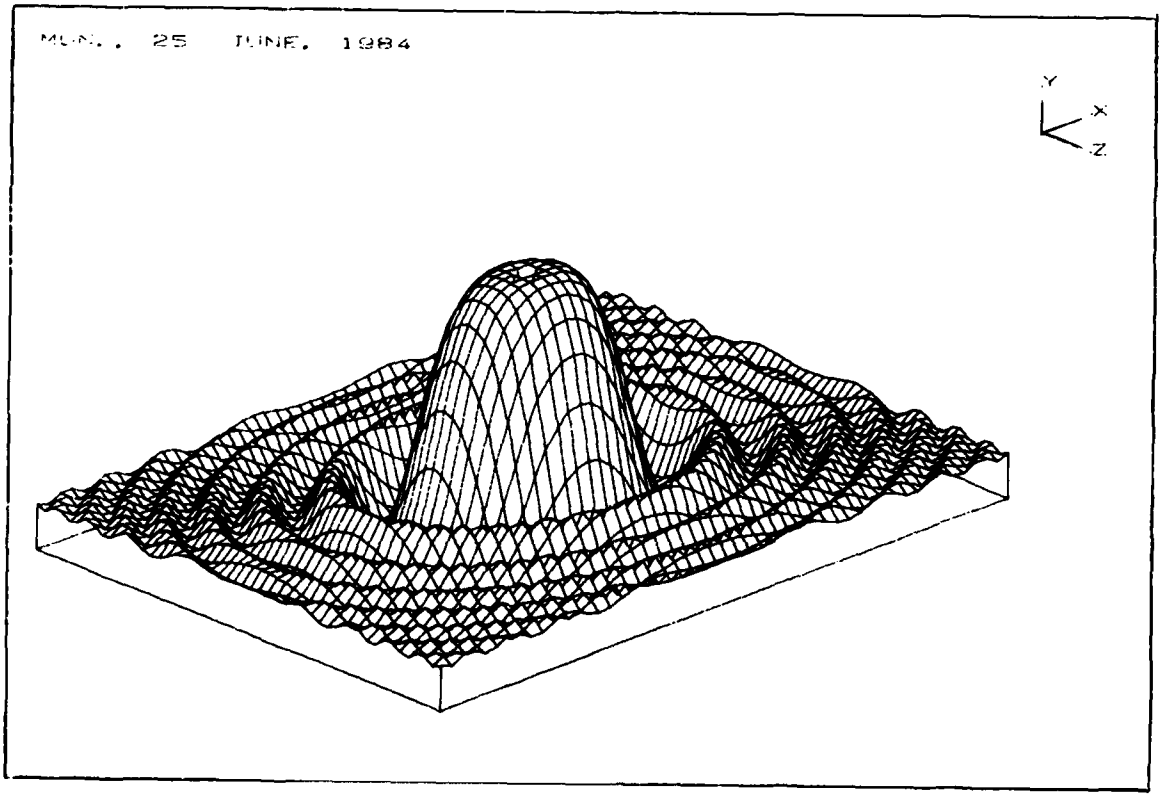


Fig. 3 — Example plot of a 3-D surface

```
Key #  1 ~ Plotting frequency of data (2)
Key #  2 ~ Number of data points
Key #  3 ~ Rotation angles of figure (35,45)
Key #  4 ~ Scaling of figure (7,2,5)
Key #  5 ~ Vertical lines for perimeter (0)
Key #  6 ~ Draw axis orientation (yes)
Key #  7 ~ Information & Title (yes,no)
Key #  8 ~ Masking on/off (no, fast plot)
Key #  9 ~ Size of figure (one, 1)
Key # 10 ~        Continue plotting
Key # 11 ~ Exit - to main program

      Select a function
 Default values in parenthesis
 Type function number Key # = i
```

Fig. 4 — Data menu for parameter input

```
*********************************************************
**** SPECIFY THE FREQUENCY OF PLOTING ON THE X-AXIS ****
**** IF IPLX = 2, THEN ONE LINE EVERY TWO POINTS ON ****
**** THE X-AXIS IS PLOTED.  IPLX =2
*********************************************************
**** SPECIFY THE FREQUENCY OF PLOTING ON THE Z-AXIS ****
**** IF IPLZ = 2, THEN ONE LINE EVERY TWO POINTS ON ****
**** THE Z-AXIS IS PLOTED.  IPLZ =2
```

Fig. 5 — Input for plotting frequency parameters

```
*********************************************************
**** SPECIFY THE NUMBER OF DATA POINTS FOR THE      ****
**** X(NXPL) ARRAY.  NXPL =101
*********************************************************
**** SPECIFY THE NUMBER OF DATA POINTS FOR THE      ****
**** Z(NZPL) ARRAY.  NZPL =101
```

Fig. 6 — Input for number of data points

```
*********************************************************
**** SPECIFY THE RELATIVE VERTICAL POSITION OF THE  ****
**** ORIGIN FOR THE AXIS.  IF DELTA = 0.0, THE ORIGIN****
**** IS CENTERED ON THE LEFT SIDE.  DELTA =0.0
*********************************************************
**** SPECIFY THE ROTATION ANGLE AROUND THE X-AXIS   ****
**** THE ROTATION IS DEFINED COUNTER-CLOCKWISE AS   ****
**** THE X-AXIS IS POINTED TOWARDS YOU ! THETA =35
*********************************************************
**** SPECIFY THE ROTATION ANGLE AROUND THE Y-AXIS   ****
**** THE ROTATION IS DEFINED COUNTER-CLOCKWISE AS   ****
**** THE Y-AXIS IS POINTED TOWARDS YOU ! PHI =45
```

Fig. 7 — Input for rotation angles about *x-y* axes

```
**********************************************************
**** SPECIFY THE SCALING IN THE X-DIRECTION.          ****
**** XSCAL0 =7.0
**********************************************************
**** SPECIFY THE SCALING IN THE Y-DIRECTION.          ****
**** YSCAL0 =2.0
**********************************************************
**** SPECIFY THE SCALING IN THE Z-DIRECTION.          ****
**** ZSCAL0 =5.0
```

Fig. 8 — Input for scaling factor

Figure 9 shows the prompts for option 5. Default values (IPERIM = 1, IPLPX = 0, IPLPZ = 0) will draw a perimeter line on the x-z plane indicating the projection of the surface plotted onto that plane. Values for IPLPX and IPLPZ other than zero (e.g. 10) would cause vertical lines to be drawn from the edge of the surface to the projection plane (every 10th data points) unless IPERIM = 0 in which case no perimeter line will be drawn and the second and third prompts of Fig. 9 are omitted.

```
**********************************************************
**** DO YOU WANT A PERIMETER DRAWN ON YOUR PLOTTED  ****
**** FIGURE ? TYPE 0 FOR NO OR 1 FOR YES  IPERIM =1
**********************************************************
**** SPECIFY THE FREQUENCY FOR DRAWING VERTICAL LINES****
**** ON THE PERIMETER  IF IPLPX=0 ONLY VERTICAL LINES****
**** AT THE CORNERS WILL BE DRAWN  (X-AXIS).IPLPX =0
**********************************************************
**** SPECIFY THE FREQUENCY FOR DRAWING VERTICAL LINES****
**** ON THE PERIMETER  IF IPLPZ=0 ONLY VERTICAL LINES****
**** AT THE CORNERS WILL BE DRAWN. (Z-AXIS). IPLPZ =0
```

Fig. 9 — Input for perimeter parameters

Figure 10 show the prompt for the reference axis display option. An example of the reference axis is shown in Fig. 3.

```
**********************************************************
**** DO YOU WANT THE REFERENCE AXIS DRAWN ON THE    ****
**** FIGURE ? TYPE 0 FOR NO OR 1 FOR YES. IAXIS =1
```

Fig. 10 — Input for reference axes

As indicated by the prompts shown in Fig. 11 (option 7) the block of information about the input data at the top of the plot and/or the title block at the bottom of the plot can be omitted by entering values of 0. Values of 1 for INFORM ans ITITLE will retain the information and title blocks. For ITITLE = 1 the user is requested to type in a figure title up to two lines (52 characters per line).

Figure 12 shows the prompt for option 8. The default value of IMASK is 0 and hidden lines are not removed. Masking of hidden lines is a very CPU intensive task and much more time is required to draw the plots if masking is requested. The increased plotting time is very noticeable both when the program is plotting to the graphics terminal or to the pen plotter. Usually masking will be selected only at the last step of developing a presentation plot.

```
**********************************************************
**** DO YOU WANT ANY INFORMATION OF THE INPUT DATA  ****
**** PLACED ON THE SAME PAGE AS THE PLOT ? INFORM =1
**********************************************************
**** DO YOU WANT THE TITLE OF THE FIGURE PLACED ON  ****
**** THE SAME PAGE ? TYPE 0/NO.1/YES. ITITLE =1
```

Fig. 11 — Input for information and title of plot

```
**********************************************************
**** SPECIFY IF HIDDEN LINE REMOVAL IS REQUIRED  IF ****
**** IMASK = 0 THEN HIDDEN LINES ARE PLOTTED  IF    ****
**** IMASK = 1 THEN HIDDEN LINES ARE NOT PLOTTED    ****
**** FOR IMASK = 0 A FAST PLOT IS DRAWN  IMASK =0
```

Fig. 12 — Input for hidden line removal

The prompt for option 9 is shown in Fig. 13. The default value of SIZE is 1.0 and the program sizes the figure to use the full available plotting area. If a value of 2.0 is specified, the plotted figure will be reduced in size one half in each direction. Normally SIZE will be set to 1.0 when plotting to the graphics terminal and to the small pen plotters such as the HP-7470. When plotting to pen plotters such as the HP-7475 and the HP-9872 a value of 1.4 or 2.0 should be used when the paper does not cover the full area of the plotter bed.

```
**********************************************************
**** SPECIFY THE SIZE OF THE PAGE THAT YOU WANT TO  ****
**** USE. IF SIZE = 1 THEN A FULL SIZE PAGE OF THE  ****
**** PLOT DEVICE IS USED  IF SIZE = 2 THEN ONLY ONE ****
**** HALF OF THE PAGE IS USED. SIZE =1.0
```

Fig. 13 — Input for figure size

Choice 10 on the display menu brings up the plotting menu so that the changed parameters can be used. This menu is shown in Fig. 14. Option 2 replots the current data with the revised parameters. The user can change as many or as few parameters between plots as is appropriate. Option 3 and 4 will read more data from the file which is currently open and, for each figure, the program will prompt for the number of points in the x and z directions. Option 3 uses the default display parameters and option 4 uses the current (or revised) display parameters. Option 1 from the plotting menu causes the program to prompt for a new file name and number of points to be read from the file. It also resets the display parameters to the default values.

Selecting option 5 on the plotting menu as well as option 11 on the display menu causes control to be returned to the main program and the prompt in Fig. 15 to be displayed. At this prompt, a response of 0 will cause the program to terminate while a response of 1 will transfer control to segment PLT3P to plot the current data on the pen plotter. If data are plotted on the pen plotter, the program prompts for pen color and pen speed as shown in Fig. 16. The pen color numbers refer to the pen stall numbers on the plotter. The plot frame and information (if requested) will be plotted with the pen in stall 1 and the data will be plotted with the pen in the stall number corresponding to the pen color selected. The program will pause for the user to ready the plotter before continuing and drawing the plot.

```
Key #  1 ~ Plot new file - default parameters
          Also old file - default parameters
Key #  2 ~ Plot same file - new parameters
          Old set of data points
Key #  3 ~ Plot same file - default parameters
          New set of data points
Key #  4 ~ Plot same file - new parameters
          New set of data points
Key #  5 ~ Exit - to main program

  Select a PLOTTING function
 Type function number Key # = 2
```

Fig. 14 — Program menu

```
##########################################
##### Do you want to plot the last     #####
##### figure on the PEN plotter ?      #####
##### IF YES type 1  if NO type 0 :1
```

Fig. 15 — Input for hard copy generation

```
**********************************************************
**** SPECIFY THE PEN COLOR FOR THE PLOTTER          ****
**** 1  BLACK, 2  BLUE, 3  RED, 4  GREEN           ****
**** PEN COLOR =1
**********************************************************
**** SPECIFY THE PEN SPEED FOR THE PLOTTER, (1-35)  ****
**** RECOMMENDED VALUE FOR QUALITY PLOTS   5        ****
**** PEN SPEED =5.0
```

Fig. 16 — Input for pen plotter parameters

At completion, program control returns to the main segement and the program prompts for continuation. Again, a response of 0 terminates the program and a response of 1 returns control to segment PLT3T so that the user can develop additional figures.

3. CODE INSTALLATION

The authors have been using the PLOT3D program on HP 1000 computers running under the RTE A.1 and RTE 6/VM operating systems. There are no known problems that would keep the program from running equally well under the RTE IVB and RTE A operating systems. The source code is device indpendent, but with the use of GRAPHICS 1000-II DGL the program is device dependent at load time. A listing of the source code is given in Appendix A. Figure 17 shows the loader command file for loading PLOT3D for use with HP-2648 terminals and a HP-9872S plotter. If different terminals and/or plotters are in use on the target system, the loader command file will have to be changed accordingly.

```
!PLT3D

0001  OP.EB
0002  EC
0003  RE,%PLT3D
0004  RE,%PLT3T
0005  SE,$A0001::VM
0006  SE,$B0001::VM
0007  SE,$K0001::VM
0008  SE,$D0001::VM
0009  SE,$DIDD::VM
0010  RE,%PLT3P
0011  SE,$A0001::VM
0012  SE,$B0001::VM
0013  SE,$K0001::VM
0014  SE,$D0003::VM
0015  SE,$DIDD::VM
0016  END
```

Fig. 17 — LOADER command file for HP-2648 and HP-9872S graphics devices

REFERENCE

1. Watkins, Steven L., "Masked Three-Dimensional Plot Program with Rotations [J6]," *Collected Algorithms From CACM*, Vol. 3, 1973.

APPENDIX

```
FTN7X,Q
$FILES 0,1
$EMA(BLK1)
      PROGRAM PLT3D(3,95), G.A.KERAMIDAS 3D:12:12:83
C*****
      COMMON /BLK1/MASK(9000),X(151),Z(151),Y(151,151)
      COMMON/DATA1/XMARG,XWMAX,XINF,YINF,XLINF,YWMAX,XYMAR,ASPR
      COMMON/DATA2/IPLX,IPLZ,THETA,PHI,DELTA,XSC,YSC,ZSC,SIZE
      COMMON/DATA3/IPLP,ISURF,INFORM,ITITLE,IPERIM,IAXIS,NFIGS,IMASK
      COMMON/DATA4/XMIN,XMAX,ZMIN,ZMAX,YMIN,YMAX
      COMMON/DATA5/NXPL,NZPL,IPLPX,IPLPZ,XSCALO,YSCALO,ZSCALO
      COMMON/DATA6/INSTN,IFAST,IPLOT,IPMENU,IPEN(2)
      COMMON/DATA7/NHEDX,HEDX(5),NHEDZ,HEDZ(5),NHEDY,HEDY(5)
      COMMON/DATA8/NTITL1,TITLE1(2),NTITL2,TITLE2(15),NTITL3,TITLE3(15)
      DIMENSION ISEG1(3),ISEG2(3)
      DIMENSION LBUF (510)
      DATA ANO/2HNO/AYE/2HYE/
       DATA ISEG1/2HPL,2HT3,2HT /
       DATA ISEG2/2HPL,2HT3,2HP /
      CALL LGBUF(LBUF,510)
C*****
C***** THIS IS A DRIVER FOR THE 3D PLT PROGRAMS.
C***** PLT3P IS THE PROGRAM FOR THE PEN PLOTTER.
C***** PLT3T IS THE PROGRAM FOR THE TERMINAL.
C*****
      IPMENU=0
      WRITE(1,1005)
 1005 FORMAT("XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
     C    ,/,"XXXXX                                                    XXXXX"
     C    ,/,"XXXXX      * * * * *   PLOT 3-D   * * * * *             XXXXX"
     C    ,/,"XXXXX                                                    XXXXX"
     C    ,/,"XXXXX This program will plot a three dimensional         XXXXX"
     C    ,/,"XXXXX surface on the Screen or on the pen Plotter.       XXXXX"
     C    ,/,"XXXXX                                                    XXXXX"
     C    ,/,"XXXXX The default values of the plotting parameters      XXXXX"
     C    ,/,"XXXXX have been set for a fast plot on the screen.       XXXXX"
     C    ,/,"XXXXX After this fast plot the user can change the       XXXXX"
     C    ,/,"XXXXX parameters for the plot desired.                   XXXXX"
     C    ,/,"XXXXX                                                    XXXXX"
     C    ,/,"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX")
    1    CONTINUE
      IMASK=0
      IPLOT=0
   11 CONTINUE
      ID=1
      CALL SEGLD(ISEG1,IERR)
  100 CONTINUE
      WRITE(1,20)
   20 FORMAT('##############################################"
     C    ,/,"##### Do you want to plot the last     #####"
     C    ,/,"##### figure on the PEN plotter ?   ")
      READ(1,1000) ANSW
 1000 FORMAT(A2)
      IF(ANSW .EQ.AYE) GO TO 21
```

```
      GO TO 300
   21 CONTINUE
      IPLOT=1
      ID = 2
      IPEN(1)=10
      IPEN(2)=1
      CALL SEGLD(ISEG2,IERR)
  200 CONTINUE
      WRITE(1,30)
   30 FORMAT("##########################################"
     C     ,/,"##### Do you want to plot any more     #####"
     C     ,/,"##### figures?  ")
      READ(1,1000) ANSW
      IF(ANSW .EQ.AYE) GO TO  1
  300 CONTINUE
      STOP
      END
```

```
FTN7X,Q
$FILES 0,1
$EMA/BLK1/
      PROGRAM PLT3T(5), G.A.KERAMIDAS 3T:04:25:84
      COMMON/BLK1/MASK(9000),XPL(151),ZPL(151),YPL(151,151)
      COMMON/DATA1/XMARG,XWMAX,XINF,YINF,XLINF,YWMAX,XYMAR,ASPR
      COMMON/DATA2/IPLX,IPLZ,THETA,PHI,DELTA,XSC,YSC,ZSC,SIZE
      COMMON/DATA3/IPLP,INFORM,ITITLE,IPERIM,IAXIS,IMASK
      COMMON/DATA4/XMIN,XMAX,ZMIN,ZMAX,YMIN,YMAX
      COMMON/DATA5/NXPL,NZPL,IPLPX,IPLPZ,XSCAL0,YSCAL0,ZSCAL0
      COMMON/DATA6/INSTN,IFAST,IPLOT,IPMENU
      COMMON/DATA7/NHEDX,HEDX(5),NHEDZ,HEDZ(5),NHEDY,HEDY(5)
      COMMON/DATA8/NTITL1,TITLE1(2),NTITL2,TITLE2(15),NTITL3,TITLE3(15)
      DIMENSION VIEW(4),IFILE(3)
      DATA IFILE/2H  ,2H  ,2H  /IDMENU/0/
      CALL DEVON
      IF (IPMENU .GE. 4) GO TO 210
   10 WRITE(1,1010)
 1010 FORMAT("*****************************************************"
     C    ,/,"**** SPECIFY THE NAME OF THE DATA FILE FROM WHICH   ****"
     C    ,/,"**** THIS PROGRAM WILL READ THE DATA TO BE PLOTTED  ****"
     C    ,/,"**** FILE NAME IFILE = ")
      READ(1,1011) IFILE
       OPEN (UNIT=5 ,FILE=IFILE )
      REWIND 5
 1011 FORMAT(3A2)
   15 WRITE(1,1015)
 1015 FORMAT("*****************************************************"
     C    ,/,"**** SPECIFY THE NUMBER OF DATA POINTS FOR THE      ****"
     C    ,/,"**** X(NXPL) ARRAY. NXPL = ")
      READ(1,*) NXPL
   20 WRITE(1,1020)
 1020 FORMAT("*****************************************************"
     C    ,/,"**** SPECIFY THE NUMBER OF DATA POINTS FOR THE      ****"
     C    ,/,"**** Z(NZPL) ARRAY. NZPL = ")
      READ(1,*) NZPL
      IF(IPMENU.GE.4) GO TO 120
  100   IPLX=2
        IPLZ=2
       DELTA=0.0
       THETA=35.0
         PHI=45.0
      XSCAL0=7
      YSCAL0=2
      ZSCAL0=5
         XSC=XSCAL0
         YSC=YSCAL0
         ZSC=ZSCAL0
       IPLPX=0
       IPLPZ=0
      IPERIM=1
       IAXIS=1
      INFORM=1
      ITITLE=1
```

```
      IMASK=0
       SIZE=1.0
      IFAST=0
       IOFF=1
        ION=2
120 CONTINUE
      READ (5) (XPL(I),I=1,NXPL)
      READ (5) (ZPL(J),J=1,NZPL)
     DO 800 I=1,NXPL
      READ (5) (YPL(I,J),J=1,NZPL)
800 CONTINUE
150 XMAX = XPL(1)
     XMIN = XMAX
     ZMAX = ZPL(1)
     ZMIN = ZMAX
     YMAX = YPL(1,1)
     YMIN = YMAX
     DO 200 I = 1,NXPL
     DO 200 J = 1,NZPL
     IF(XPL(I).LT.XMIN) XMIN = XPL(I)
     IF(XPL(I).GT.XMAX) XMAX = XPL(I)
     IF(ZPL(J).LT.ZMIN) ZMIN = ZPL(J)
     IF(ZPL(J).GT.ZMAX) ZMAX = ZPL(J)
     IF(YPL(I,J).LT.YMIN)  YMIN = YPL(I,J)
     IF(YPL(I,J).GT.YMAX)  YMAX = YPL(I,J)
200 CONTINUE
       XMARG=0.25/SIZE
       XYMAR=XMARG
        ASPR=1.0
       XWMAX=10./SIZE
       YWMAX=10./SIZE
          X0=0.0
          Y0=0.0
      XPMAX=0.0
      YPMAX=0.0
       XINF=XMARG
       YINF=YWMAX-XMARG
      XLINF=1.0/SIZE
      YLINF=1./SIZE
210  CONTINUE
     CALL VIEWP(XWMAX,YWMAX,SIZE,IERROR)
     CALL ZIWS(451,0.4,IDUM,VIEW,IERROR)
     IF(IERROR.EQ.0) GO TO 305
     CALL ERRORS(1,IERROR,10HZIWS. 451 )
     GO TO 999
305 IF(INFORM.LE.0) GO TO 310
       Y1=0.8*YWMAX
      YPMAX=.2*VIEW(4)
     CALL PLTIF(X0,XWMAX,Y1,YWMAX)
310 IF(ITITLE.LE.0) GO TO 315
       Y2=0.08*YWMAX
     CALL TITLE(X0,XWMAX,Y0,Y2 )
       Y0=.08*VIEW(4)
315  CONTINUE
```

```
      CALL ZIWS(451,0,4,IDUM,VIEW,IERROR)
      IF(IERROR.EQ.0) GO TO 400
      CALL ERRORS(1,IERROR,10HZIWS. 451 )
      GO TO 999
  400    XV0=VIEW(1)
         YV0=VIEW(3)+YG
        XVMAX=VIEW(2)
        YVMAX=VIEW(4)-YPMAX
  410 CALL ZVIEW(XV0,XVMAX,YV0,YVMAX)
      CALL ADISP
      CALL BOXIN(.0,.0,XWMAX,YWMAX,0.0,0.0)
      CALL PLOTS(MASK,XPL,ZPL,YPL,NXPL,NZPL,1,NXPL)
      CALL ZMCUR
      WRITE(1,2000)
      READ(1,*) ICONT
      CALL GDISP(IOFF)
      CALL DMENU(IDMENU)
      IPMENU=5
      IF (IDMENU .EQ. 11) GO TO 999
      CALL PMENU(IPMENU)
      CALL GDISP(ION)
      GO TO (10,150,15,15,500), IPMENU
  500 CONTINUE
 2000 FORMAT (/////////////////////////,
     *        "  Type ( / ) to continue when ready : _")
  999 CONTINUE
      CALL DEVOF
      CALL SEGRT (Z99)
      END
      SUBROUTINE PMENU(IPMENU)
      CALL PFUNCT
  100 CALL ZALPH(30,30H   Select a PLOTTING function     )
      WRITE(1,1000)
 1000 FORMAT("  Type function number Key # =  ")
      READ(1,*) IPMENU
      IF(IPMENU.LT.1 .OR.IPMENU.GT. 5) GO TO 100
      RETURN
      END
      SUBROUTINE PFUNCT
      CALL ADISP
      CALL ZNEWF
      CALL ZALPH(48,48H Key #  1 ~ Plot new file - default parameters  )
      CALL ZALPH(48,48H            Also old file - default parameters  )
      CALL ZALPH(48,48H Key #  2 ~ Plot same file - new parameters     )
      CALL ZALPH(48,48H            Old set of data points             )
      CALL ZALPH(48,48H Key #  3 ~ Plot same file - default parameters )
      CALL ZALPH(48,48H            New set of data points             )
      CALL ZALPH(48,48H Key #  4 ~ Plot same file - new parameters     )
      CALL ZALPH(48,48H            New set of data points             )
      CALL ZALPH(48,48H Key #  5 ~ Exit - to main program             )
      CALL ZALPH(1 , 1H                                               )
      RETURN
      END
      SUBROUTINE DMENU(IDMENU)
```

```
  10 CONTINUE
     CALL DFUNCT
 100 CALL ZALPH(30,30H        Select a function      )
     CALL ZALPH(40,40H  Default values in papenthesis            )
     WRITE(1,1000)
1000 FORMAT("  Type function number Key # = ..")
     READ(1,*) IDMENU
     IF(IDMENU.LT.1 .OR.IDMENU.GT.11) GO TO 100
     IF(IDMENU.GE.10) GO TO 999
     CALL PLPAR(IDMENU)
     GO TO 10
 999 CONTINUE
     RETURN
     END
     SUBROUTINE DFUNCT
     COMMON/DATA2/IPLX,IPLZ,THETA,PHI,DELTA,XSC,YSC,ZSC,SIZE
     COMMON/DATA3/IPLP,INFORM,ITITLE,IPERIM,IAXIS,IMASK
     COMMON/DATA5/NXPL,NZPL,IPLPX,IPLPZ,XSCAL0,YSCAL0,ZSCAL0
     CALL ADISP
     CALL ZALPH(45,45H Key #  1 ~ Plotting frequency of data (2)      )
     CALL ZALPH(45,45H Key #  2 ~ Number of data points               )
     CALL ZALPH(45,45H Key #  3 ~ Rotation angles of figure (35,45)   )
     CALL ZALPH(45,45H Key #  4 ~ Scaling of figure (7.2.5)           )
     CALL ZALPH(45,45H Key #  5 ~ Vertical lines for perimeter (0)    )
     CALL ZALPH(45,45H Key #  6 ~ Draw axis orientation (yes)          )
     CALL ZALPH(45,45H Key #  7 ~ Information & Title (yes,no)        )
     CALL ZALPH(45,45H Key #  8 ~ Masking on/off (no, fast plot)      )
     CALL ZALPH(45,45H Key #  9 ~ Size of figure (one, 1)             )
     CALL ZALPH(45,45H Key # 10 ~        Continue plotting            )
     CALL ZALPH(45,45H Key # 11 ~ Exit - to main program              )
     CALL ZALPH(1 , 1H                                                )
     RETURN
     END
     SUBROUTINE ADISP
     INTEGER ILIST(7), STRING(2), IERR
     DATA STRING /15550B,15512B/
C
     CALL ZALPH (4,STRING)
C
     RETURN
     END
C
     SUBROUTINE GDISP(IGRF)
     DIMENSION ILIST(2)
C
     GO TO (10,20),IGRF
  10 ILIST(1)=0
     CALL ZOESC(1055,1,0,ILIST,DUMMY,IERROR)
     GO TO 9999
  20 ILIST(1)=1
     CALL ZOESC(1055,1,0,ILIST,DUMMY,IERROR)
C
9999 RETURN
     END
```

```
C
      SUBROUTINE PLPAR(IDMENU)
      COMMON/DATA1/XMARG,XWMAX,XINF,YINF,XLINF,YWMAX,XYMAR,ASPR
      COMMON/DATA2/IPLX,IPLZ,THETA,PHI,DELTA,XSC,YSC,ZSC,SIZE
      COMMON/DATA3/IPLP,INFORM,ITITLE,IPERIM,IAXIS,IMASK
      COMMON/DATA4/XMIN,XMAX,ZMIN,ZMAX,YMIN,YMAX
      COMMON/DATA5/NXPL,NZPL,IPLPX,IPLPZ,XSCAL0,YSCAL0,ZSCAL0
      GOTO (5,15,25,40,55,65,70,80,85), IDMENU
    5 WRITE(1,1005)
 1005 FORMAT("*************************************************************"
     C    ,/,"**** SPECIFY THE FREQUENCY OF PLOTING ON THE X-AXIS ****"
     C    ,/,"**** IF IPLX = 2, THEN ONE LINE EVERY TWO POINTS ON ****"
     C    ,/,"**** THE X-AXIS IS PLOTED. IPLX =_")
      READ(1,*) IPLX
   10 WRITE(1,1010)
 1010 FORMAT("*************************************************************"
     C    ,/,"**** SPECIFY THE FREQUENCY OF PLOTING ON THE Z-AXIS ****"
     C    ,/,"**** IF IPLZ = 2, THEN ONE LINE EVERY TWO POINTS ON ****"
     C    ,/,"**** THE Z-AXIS IS PLOTED  IPLZ =_")
      READ(1,*) IPLZ
      GO TO 2000
   15 WRITE(1,1015)
 1015 FORMAT("*************************************************************"
     C    ,/,"**** SPECIFY THE NUMBER OF DATA POINTS FOR THE      ****"
     C    ,/,"**** X(NXPL) ARRAY. NXPL =_")
      READ(1,*) NXPL
   20 WRITE(1,1020)
 1020 FORMAT("*************************************************************"
     C    ,/,"**** SPECIFY THE NUMBER OF DATA POINTS FOR THE      ****"
     C    ,/,"**** Z(NZPL) ARRAY. NZPL = ")
      READ(1,*) NZPL
      GO TO 2000
   25 WRITE(1,1025)
 1025 FORMAT("*************************************************************"
     C    ,/,"**** SPECIFY THE RELATIVE VERTICAL POSITION OF THE  ****"
     C    ,/,"**** ORIGIN FOR THE AXIS. IF DELTA = 0.0, THE ORIGIN****"
     C    ,/,"**** IS CENTERED ON THE LEFT SIDE  DELTA = ")
      READ(1,*) DELTA
   30 WRITE(1,1030)
 1030 FORMAT("*************************************************************"
     C    ,/,"**** SPECIFY THE ROTATION ANGLE AROUND THE X-AXIS.  ****"
     C    ,/,"**** THE ROTATION IS DEFINED COUNTER-CLOCKWISE AS   ****"
     C    ,/,"**** THE X-AXIS IS POINTED TOWARDS YOU ! THETA = ")
      READ(1,*) THETA
   35 WRITE(1,1035)
 1035 FORMAT("*************************************************************"
     C    ,/,"**** SPECIFY THE ROTATION ANGLE AROUND THE Y-AXIS.  ****"
     C    ,/,"**** THE ROTATION IS DEFINED COUNTER-CLOCKWISE AS   ****"
     C    ,/,"**** THE Y-AXIS IS POINTED TOWARDS YOU ! PHI =_")
      READ(1,*) PHI
      GO TO 2000
   40 WRITE(1,1040)
 1040 FORMAT("*************************************************************"
     C    ,/,"**** SPECIFY THE SCALING IN THE X-DIRECTION.        ****"
```

```
     C    ,/,"**** XSCAL0 = ")
          READ(1,*) XSCAL0
      45  WRITE(1,1045)
    1045  FORMAT("*************************************************************"
     C    ,/,"**** SPECIFY THE SCALING IN THE Y-DIRECTION.        ****"
     C    ,/,"**** YSCAL0 = ")
          READ(1,*) YSCAL0
      50  WRITE(1,1050)
    1050  FORMAT("*************************************************************"
     C    ,/,"**** SPECIFY THE SCALING IN THE Z-DIRECTION.        ****"
     C    ,/,"**** ZSCAL0 = ")
          READ(1,*) ZSCAL0
          GO TO 2000
      55  WRITE(1,1055)
    1055  FORMAT("*************************************************************"
     C    ,/,"**** DO YOU WANT A PERIMETER DRAWN ON YOUR PLOTTED  ****"
     C    ,/,"**** FIGURE ? TYPE 0 FOR NO OR 1 FOR YES. IPERIM = ")
          READ(1,*) IPERIM
          IF(IPERIM.LE.0) GO TO 65
      60  WRITE(1,1060)
    1060  FORMAT("*************************************************************"
     C    ,/,"**** SPECIFY THE FREQUENCY FOR DRAWING VERTICAL LINES***"
     C    ,/,"**** ON THE PERIMETER. IF IPLPX=0 ONLY VERTICAL LINES***"
     C    ,/,"**** AT THE CORNERS WILL BE DRAWN. (X-AXIS),IPLPX =")
          READ(1,*) IPLPX
          WRITE(1,1061)
    1061  FORMAT("*************************************************************"
     C    ,/,"**** SPECIFY THE FREQUENCY FOR DRAWING VERTICAL LINES***"
     C    ,/,"**** ON THE PERIMETER. IF IPLPZ=0 ONLY VERTICAL LINES***"
     C    ,/,"**** AT THE CORNERS WILL BE DRAWN. (Z-AXIS), IPLPZ =")
          READ(1,*) IPLPZ
          GO TO 2000
      65  WRITE(1,1065)
    1065  FORMAT("*************************************************************"
     C    ,/,"**** DO YOU WANT THE REFERENCE AXIS DRAWN ON THE    ****"
     C    ,/,"**** FIGURE ? TYPE 0 FOR NO OR 1 FOR YES. IAXIS = ")
          READ(1,*) IAXIS
          GO TO 2000
      70  WRITE(1,1070)
    1070  FORMAT("*************************************************************"
     C    ,/,"**** DO YOU WANT ANY INFORMATION OF THE INPUT DATA  ****"
     C    ,/,"**** PLACED ON THE SAME PAGE AS THE PLOT ? INFORM = ")
          READ(1,*) INFORM
      75  WRITE(1,1075)
    1075  FORMAT("*************************************************************"
     C    ,/,"**** DO YOU WANT THE TITLE OF THE FIGURE PLACED ON  ****"
     C    ,/,"**** THE SAME PAGE ? TYPE 0/NO,1/YES, ITITLE = ")
          READ(1,*) ITITLE
          GO TO 2000
      80  WRITE(1,1080)
    1080  FORMAT("*************************************************************"
     C    ,/,"**** SPECIFY IF HIDDEN LINE REMOVAL IS REQUIRED. IF ****"
     C    ,/,"**** IMASK = 0 THEN HIDDEN LINES ARE PLOTTED. IF    ****"
     C    ,/,"**** IMASK = 1 THEN HIDDEN LINES ARE NOT PLOTTED.   ****"
```

```
C     ,/,"**** FOR IMASK = 0 A FAST PLOT IS DRAWN. IMASK = ")
      READ(1,*) IMASK
      GO TO 2000
   85 WRITE(1,1090)
 1090 FORMAT("************************************************************"
     C     ,/,"**** SPECIFY THE SIZE OF THE PAGE THAT YOU WANT TO  ****"
     C     ,/,"**** USE. IF SIZE = 1 THEN A FULL SIZE PAGE OF THE  ****"
     C     ,/,"**** PLOT DEVICE IS USED. IF SIZE = 2 THEN ONLY ONE ****"
     C     ,/,"**** HALF OF THE PAGE IS USED. SIZE = ")
      READ(1,*) SIZE
      IF(SIZE.LE.0.0) SIZE=1.0
 2000 CONTINUE
          XSC=XSCAL0/SIZE
          YSC=YSCAL0/SIZE
          ZSC=ZSCAL0/SIZE
      RETURN
      END
      SUBROUTINE PLOTS(MASK,X,Z,Y,NX,NZ,IDIM,IDIMEN)
      LOGICAL LPLOTX,LPLOTZ
      DIMENSION XX(151),ZZ(151),YY(151),VERTEX(16)
      EMA MASK(9000),X(151),Z(151),Y(151,151)
      COMMON/DATA1/XMARG,XWMAX,XINF,YINF,XLINF,YWMAX,XYMAR,ASPR
      COMMON/DATA2/IPLX,IPLZ,THETA,PHI,DELTA,XSC,YSC,ZSC,SIZE
      COMMON/DATA3/IPLP,INFORM,ITITLE,IPERIM,IAXIS,IMASK
      COMMON/DATA5/NXPL,NZPL,IPLPX,IPLPZ,XSCAL0,YSCAL0,ZSCAL0
      COMMON/DATA4/XMIN,XMAX,ZMIN,ZMAX,YMIN,YMAX
      DATA SW/-1./
      PID180= 01745329252
      PIR=3.14159
      COSPH = COS( PID180 * PHI )
      COSTH = COS( PID180 * THETA )
      SINPH = SIN( PID180 * PHI )
      SINTH = SIN( PID180 * THETA )
      TMP1 = XSC * COSPH
      TMP2 = ZSC * SINPH
      XMX  = AMAX1( TMP1 , TMP2 , TMP1+TMP2 , 0. )
      XMN  = AMIN1( TMP1 , TMP2 , TMP1+TMP2 , 0. )
      TMP1 = XSC * SINPH *SINTH
      TMP2 = -ZSC *COSPH *SINTH
      YMX  = AMAX1( TMP1 , TMP2 , TMP1+TMP2 , 0  )
      YMN  = AMIN1( TMP1 , TMP2 , TMP1+TMP2 , 0. )
      TMP1 = YSC * COSTH
      YMX  = AMAX1( YMX , YMX + TMP1 )
      YMN  = AMIN1( YMN , YMN + TMP1 )
      XL = XMX - XMN
      YL = YMX - YMN
      IF( XL .GT. XWMAX  .OR.  YL .GT. YWMAX ) GO TO 998
      XOR = XMARG - XMN
      XMX = XMARG + XL + XMARG
      SW=-SW
      YOR = ABS(YMN)+(YWMAX-YL)/2.+DELTA
      LPLOTX = -SINPH * COSTH .GT. 0.
      LPLOTZ =  COSPH * COSTH .GT. 0.
      XSCALE = XSC/(XMAX-XMIN)
```

```
      YSCALE = YSC/(YMAX-YMIN)
      ZSCALE = ZSC/(ZMAX-ZMIN)
      CALL ZMOVE( XOR , 0.0)
      KPLX = MAX0( IPLX , 1 )
      NEXTX = 1
      DO 450 NLINE = 1,NX
      IF( NLINE .NE. NEXTX ) GO TO 450
      NEXTX = MIN0( NLINE + KPLX , NX )
      NN = NLINE
      IF( LPLOTX ) NN = NX + 1 - NLINE
      DO 400 J=1 , NZ
      XX(J) = X(NN)-XMIN
      YY(J) = Y(NN,J)-YMIN
      ZZ(J) = Z(J)-ZMIN
  400 CONTINUE
      NN = NLINE
      CALL PLOTF( 1111,XX,YY,ZZ,XSCALE,YSCALE,ZSCALE,NN,NZ,-PHI,-THETA,
     *          XOR,YOR,XMX,MASK,VERTEX)
  450 CONTINUE
      IF(IPERIM.GE.1)
     *CALL PERIM(X,Y,Z,XX,YY,ZZ,XSCALE,YSCALE,ZSCALE,XMIN,YMIN,ZMIN,
     1NX,NZ,PHI,THETA,XOR,YOR,XMX,MASK,NN,1)
      KPLZ = MAX0( IPLZ , 1 )
      NEXTZ = 1
      DO 550 NLINE = 1 , NZ
      IF( NLINE .NE. NEXTZ ) GO TO 550
      NEXTZ = MIN0( NLINE + KPLZ , NZ )
      NN = NLINE
      IF( LPLOTZ ) NN = NZ + 1 - NLINE
      DO 500  I = 1 , NX
      XX(I) = X(I)-XMIN
      YY(I) = Y(I,NN)-YMIN
      ZZ(I) = Z(NN)-ZMIN
  500 CONTINUE
      NN = NLINE
      CALL PLOTF( 0111,XX,YY,ZZ,XSCALE,YSCALE,ZSCALE,NN,NX,-PHI,-THETA,
     *           XOR,YOR,XMX,MASK,0.0)
  550 CONTINUE
      IF(IPERIM.GE.1)
     *CALL PERIM(X,Y,Z,XX,YY,ZZ,XSCALE,YSCALE,ZSCALE,XMIN,YMIN,ZMIN,
     1NX,NZ,PHI,THETA,XOR,YOR,XMX,MASK,NN,2)
         XC0=XWMAX-4.*XMARG
         YC0=.85*YWMAX
         ZC0=XC0
        XLMX=2.5*XMARG
        YLMX=3.0*XMARG
        ZLMX=2.5*XMARG
      IF(IAXIS.GE.1)   CALL REFAX(XC0,YC0,ZC0,XLMX,YLMX,ZLMX)
       XP0=XMARG
       YP0=.93*YWMAX
      CALL DATEP(XP0,YP0)
      GO TO 999
  998 CONTINUE
      CALL LETTR(0.2,0.25,.02,26HFIGURE SIZE LIMIT EXCEEDED,0.,26 )
```

```
999  CONTINUE
     RETURN
     END
     SUBROUTINE PLOTF(IVXYZ,XDATA,YDATA,ZDATA,XSCALE,
    *YSCALE,ZSCALE,NLINE,NPNTS,PHI,THETA,XREF,
    *YREF,XLENTH,MASK,VERTEX)
     INTEGER HIGH, OLDHI, OLDLOW
     COMMON/DATA2/IPLX0,IPLZ0,THETA0,PHI0,DELTA0,XSC0,YSC0,ZSC0,SIZE
     COMMON/DATA3/IPLP,INFORM,ITITLE,IPERIM,IAXIS,IMASK
     COMMON/DATA5/NXPL,NZPL,IPLPX,IPLPZ,XSCAL0,YSCAL0,ZSCAL0
     DIMENSION XDATA(1), YDATA(1), ZDATA(1),VERTEX(16)
     EMA MASK(9000)
     DATA INIT, JVXYZ, SPHI, STHETA/-1,-1,-1.0E30,-1.0E30/
     IF (NLINE.EQ.0) GO TO 550
     IF (NLINE.NE.1) GO TO 20
     PIPI=100.*SQRT(SIZE)
     NYPI=PIPI*20
     LIMITX=(XLENTH*PIPI)+0.5
     I=LIMITX+LIMITX
     DO 10 K=1,I
     MASK(K)=INIT
10   CONTINUE
     INIT=-1
     INCI=-1
     I=0
  20 IF(JVXYZ.EQ.IVXYZ)GO TO 70
     JVXYZ=IVXYZ
     INDZ=1
     INDY=1
     INDX=1
     INDV=1
     IF(JVXYZ.LT.1000)GO TO 30
     INDV=2
     JVXYZ=JVXYZ-1000
  30 IF(JVXYZ.LT.100)GO TO 40
     INDX=2
     JVXYZ=JVXYZ-100
  40 IF(JVXYZ.LT.10)GO TO 50
     INDY=2
     JVXYZ=JVXYZ-10
  50 IF(JVXYZ.LT.1)GO TO 60
     INDZ=2
  60 JVXYZ=IVXYZ
  70 IF (PHI.EQ.SPHI.AND.THETA.EQ.STHETA) GO TO 80
     SPHI=SIN(.0174532925*PHI)
     CPHI=COS(.0174532925*PHI)
     STHETA=SIN(.0174532925*THETA)
     CTHETA=COS(.0174532925*THETA)
     A11=CPHI
     A13=-SPHI
     A21=STHETA*SPHI
     A22=CTHETA
     A23=STHETA*CPHI
     SPHI=PHI
```

```
      STHETA=THETA
   80 INCI=-INCI
      IF(I.NE.0)I=NPNTS+1
      DO 530 K=1,NPNTS
        I=I+INCI
        GO TO (90,100),INDX
   90 X=XDATA(1)+(I-1)*XSCALE
        GO TO 110
  100   X=XDATA(I)*XSCALE
  110   GO TO (120,130),INDY
  120 Y=YDATA(1)+(I-1)*YSCALE
        GO TO 140
  130   Y=YDATA(I)*YSCALE
  140   GO TO (150,160),INDZ
  150 Z=ZDATA(1)+(NLINE-1)*ZSCALE
        GO TO 170
  160   Z=ZDATA(I)*ZSCALE
  170   XXX=A11*X+A13*Z+XREF
        XX=XXX
      IX=(XX*PIPI)+.5
        YYY=A21*X + A23*Z + YREF
        YY=YYY+A22*Y
      IY=(YY*PIPI)+.5
        IF (IX.LE.0)IX=1
        IF(IX.GT.LIMITX)IX=LIMITX
        IF(IY.LT.5)IY=5
        IF(IY.GT.NYPI)IY=NYPI
        IF (K.NE.1)GO TO 250
        LOW=IX+IX
        HIGH =LOW-1
        MLOW=MASK(LOW)
        MHIGH=MASK(HIGH)
        IF(MHIGH-IY)200,210,180
  180   IF(MLOW-IY)190,230,220
  190   LOCOLD=0
        GOTO 240
  200   MASK(HIGH)=IY
        IF(MLOW.EQ.-1)MASK(LOW)=IY
  210   LOCOLD=+1
        GO TO 240
  220   MASK(LOW)=IY
  230   LOCOLD=-1
  240   IF(IMASK) 241,241,242
  241   CALL ZMOVE(XX,YY)
        GO TO 243
  242  IF(LOCOLD.NE.0) CALL ZMOVE(FLOAT(IX)/PIPI,FLOAT(IY)/PIPI)
  243   CONTINUE
        JX=IX
        JY=IY
        IYREF=IY
        IF(INDV.EQ.1)GO TO 530
        INDEX=INCI+6
        VERTEX(INDEX)=XX
        VERTEX(INDEX+1)=YY
```

```
      VERTEX(INDEX+8)=XXX
      VERTEX(INDEX+9)=YYY
      IF(NLINE.NE.1) GO TO 530
      VERTEX(1)=XX
      VERTEX(2)=YY
      VERTEX(9)=XXX
      VERTEX(10)=YYY
      GO TO 530
 250  IF(IMASK.GT.0) GO TO 255
      CALL ZDRAW(XX,YY)
      GO TO 530
 255  IF(IX.NE.JX)GO TO 260
      JY=IY
      GO TO 280
 260  YINC=FLOAT(IY-JY)/ABS(FLOAT(IX-JX))
      INCX=(IX-JX)/IABS(IX-JX)
      YJ=JY
 270  JX=JX+INCX
      YJ=YJ+YINC
     JY=YJ+.5
      LOW=JX+JX
      HIGH=LOW-1
      MLOW=MASK(LOW)
      MHIGH=MASK(HIGH)
280   IF(MHIGH-JY)300,300,290
290   IF(MLOW-JY)310,320,320
 300  LOC=+1
      IF(LOCOLD)360,370,430
 310  LOC=0
      IF(LOCOLD)340,350,330
 320  LOC=-1
      IF(LOCOLD)510,450,440
 330 IF(MHIGH.LE.IYREF)CALL ZDRAW(FLOAT(JX)/PIPI,FLOAT(MHIGH)/PIPI)
      GO TO 350
 340 IF(MLOW.GE.IYREF)CALL ZDRAW(FLOAT(JX)/PIPI,FLOAT(MLOW)/PIPI)
 350 CONTINUE
      GO TO 520
 360  IF(MLOW-IYREF)370,380,380
 370  IF(MHIGH-IYREF)400,390,390
 380 CALL ZDRAW(FLOAT(JX)/PIPI,FLOAT(MLOW)/PIPI)
 390 CALL ZMOVE(FLOAT(JX)/PIPI,FLOAT(MHIGH)/PIPI)
      GO TO 430
 400  IF(MHIGH.EQ.-1)GO TO 430
      OLDHI=HIGH-2*INCX
      IF(MASK(OLDHI)-JY)420,420,410
 410 CALL ZMOVE(FLOAT(JX)/PIPI,FLOAT(JY)/PIPI)
      GO TO 430
420  IP=JX-INCX
     ID=MASK(OLDHI)
     CALL ZMOVE(FLOAT(IP)/PIPI,FLOAT(ID)/PIPI)
 430  MASK(HIGH)=JY
      IF(MLOW.EQ.-1)MASK(LOW)=JY
     CALL ZDRAW(FLOAT(JX)/PIPI,FLOAT(JY)/PIPI)
      GO TO 520
```

```
  440   IF(MHIGH-IYREF)460,460,450
  450   IF(MLOW-IYREF)470,470,480
  460 CALL ZDRAW(FLOAT(JX)/PIPI,FLOAT(MHIGH)/PIPI)
  470 CALL ZMOVE(FLOAT(JX)/PIPI,FLOAT(MLOW)/PIPI)
      GO TO 510
  480   OLDLOW=LOW-2*INCY
      IF(MASK(OLDLOW)-JY)490,500,500
  490 CALL ZMOVE(FLOAT(JX)/PIPI,FLOAT(JY)/PIPI)
      GO TO 510
  500  IP=JX-INCX
      ID=MASK(OLDLOW)
      CALL ZMOVE(FLOAT(IP)/PIPI,FLOAT(ID)/PIPI)
  510   MASK(LOW)=JY
      CALL ZDRAW(FLOAT(JX)/PIPI,FLOAT(JY)/PIPI)
  520   IYREF=JY
      LOCOLD=LOC
      IF(JX.NE.IX) GO TO 270
  530 CONTINUE
      IF(IMASK) 531,531,532
  531 CALL ZMOVE(XX,YY)
      GO TO 535
  532 CALL ZMOVE(FLOAT(JX)/PIPI,FLOAT(JY)/PIPI)
  535 IF(INDV.EQ.1)GO TO 540
      INDEX=-INCI+6
      VERTEX(INDEX)=XX
      VERTEX(INDEX+1)=YY
      VERTEX(INDEX+8)=XXX
      VERTEX(INDEX+9)=YYY
      IF (NLINE.NE.1)GO TO 540
      VERTEX(3)=XX
      VERTEX(4)=YY
      VERTEX(11)=XXX
      VERTEX(12)=YYY
  540 I=I-1
      RETURN
  550 INIT=0
      RETURN
      END
      SUBROUTINE PERIM(X,Y,Z,XX,YY,ZZ,XSCALE,YSCALE,ZSCALE,XMIN,YMIN,
     *                 ZMIN,NX,NZ,PHI,THETA,XOR,YOR,XMX,MASK,NN,IDIM)
      COMMON/DATA3/IPLP,INFORM,ITITLE,IPERIM,IAXIS,IMASK
      COMMON/DATA5/NXPL,NZPL,IPLPX,IPLPZ,XSCALE,YSCALE,ZSCALE
      DIMENSION XX(1),YY(1),ZZ(1)
      DIMENSION XV(2),YV(2),ZV(2)
      EMA MASK(9000),X(151),Z(151),Y(151,151)
      IDIREC=-1
      IF(IPLPX.GT.0) GO TO 2
      IPLPX=NX-1
      IPLPZ=NZ-1
    2 GO TO (4,11),IDIM
    4 DO 5 J=1,NZ
      XX(J)=X(1)-XMIN
      YY(J)=0.0
      ZZ(J)=Z(J)-ZMIN
```

```
  5 CONTINUE
    DO 10 J=1,NZ,IPLPZ
    IDIREC=-IDIREC
    XV(1)=XX(J)
    XV(2)=XV(1)
    YV(1)=0.0
    YV(2)=0.0
    IF(IDIREC.EQ.-1) YV(1)=Y(1,J)-YMIN
    IF(IDIREC.EQ.+1) YV(2)=Y(1,J)-YMIN
    ZV(1)=ZZ(J)
    ZV(2)=ZV(1)
10  CALL PLOTF(0111,XV,YV,ZV,XSCALE,YSCALE,ZSCALE,NN,2,-PHI,
   * -THETA,XOR,YOR,XMX,MASK,0.0)
    CALL PLOTF(0111,XX,YY,ZZ,XSCALE,YSCALE,ZSCALE,NN,NZ,-PHI,
   * -THETA,XOR,YOR,XMX,MASK,0.0)
    GO TO 21
11  DO 15 I=1,NX
    XX(I)=X(I)-XMIN
    YY(I)=0.0
    ZZ(I)=Z(NZ)-ZMIN
15  CONTINUE
    DO 20 I=1,NX,IPLPX
    IDIREC=-IDIREC
    XV(1)=XX(I)
    XV(2)=XV(1)
    YV(1)=0.0
    YV(2)=0.0
    IF(IDIREC.EQ.-1) YV(1)=Y(I,NZ)-YMIN
    IF(IDIREC.EQ.+1) YV(2)=Y(I,NZ)-YMIN
    ZV(1)=ZZ(I)
    ZV(2)=ZV(1)
20  CALL PLOTF(0111,XV,YV,ZV,XSCALE,YSCALE,ZSCALE,NN,2,-PHI,
   * -THETA,XOR,YOR,XMX,MASK,0.0)
    CALL PLOTF(0111,XX,YY,ZZ,XSCALE,YSCALE,ZSCALE,NN,NX,-PHI,
   * -THETA,XOR,YOR,XMX,MASK,0.0)
    GO TO 31
21  DO 25 J=1,NZ
    JJ=NZ+1-J
    XX(J)=X(NX)-XMIN
    YY(J)=0.0
    ZZ(J)=Z(JJ)-ZMIN
25  CONTINUE
    DO 30 J=1,NZ,IPLPZ
    IDIREC=-IDIREC
    JJ=NZ+1-J
    XV(1)=XX(J)
    XV(2)=XV(1)
    YV(1)=0.0
    YV(2)=0.0
    IF(IDIREC.EQ.-1) YV(1)=Y(NX,JJ)-YMIN
    IF(IDIREC.EQ.+1) YV(2)=Y(NX,JJ)-YMIN
    ZV(1)=ZZ(J)
    ZV(2)=ZV(1)
30  CALL PLOTF(0111,XV,YV,ZV,XSCALE,YSCALE,ZSCALE,NN,2,-PHI
```

```
     * -THETA,XOR,YOR,XMX,MASK,0.0)
      CALL PLOTF(0111,XX,YY,ZZ,XSCALE,YSCALE,ZSCALE,NN,NZ,-PHI,
     * -THETA,XOR,YOR,XMX,MASK,0.0)
      GO TO 41
   71 DO 35 I=1,NX
      II=NX+1-I
      XX(I)=X(II)-XMIN
      YY(I)=0.0
      ZZ(I)=Z(1)-ZMIN
   35 CONTINUE
      DO 40 I=1,NX,IPLPX
      II=NX+1-I
      IDIREC=-IDIREC
      XV(1)=XX(I)
      XV(2)=XV(1)
      YV(1)=0.0
      YV(2)=YV(1)
      IF(IDIREC.EQ.-1) YV(1)=Y(II,1)-YMIN
      IF(IDIREC.EQ.+1) YV(2)=Y(II,1)-YMIN
      ZV(1)=ZZ(I)
      ZV(2)=ZV(1)
   40 CALL PLOTF(0111,XV,YV,ZV,XSCALE,YSCALE,ZSCALE,NN,2,-PHI,
     * -THETA,XOR,YOR,XMX,MASK,0.0)
      CALL PLOTF(0111,XX,YY,ZZ,XSCALE,YSCALE,ZSCALE,NN,NX,-PHI,
     * -THETA,XOR,YOR,XMX,MASK,0.0)
   41 CONTINUE
  100 RETURN
      END
      BLOCK DATA
      COMMON/DATA1/XMARG,XWMAX,XINF,YINF,XLINF,YWMAX,XYMAR,ASPR
      COMMON/DATA2/IPLX,IPLZ,THETA,PHI,DELTA,XSC,YSC,ZSC,SIZE
      COMMON/DATA3/IPLP,INFORM,ITITLE,IPERIM,IAXIS,IMASK
      COMMON/DATA4/XMIN,XMAX,ZMIN,ZMAX,YMIN,YMAX
      COMMON/DATA5/NXPL,NZPL,IPLPX,IPLPZ,XSCAL0,YSCAL0,ZSCAL0
      COMMON/DATA7/NHEDX,HEDX(5),NHEDZ,HEDZ(5),NHEDY,HEDY(5)
      COMMON/DATA8/NTITL1,TITLE1(2),NTITL2,TITLE2(15),NTITL3,TITLE3(15)
      DATA HEDX/4H    ,4H    ,4H    ,4H    ,4H    /
      DATA HEDY/4H    ,4H    ,4H    ,4H    ,4H    /
      DATA HEDZ/4H    ,4H,   ,4H    ,4H    ,4H    /
      DATA TITLE1/4HFIGU,4HRE  /
      DATA TITLE2  /4H    ,4H    ,4H    ,4H    ,4H    ,4H    ,4H    ,
     C4H    ,4H    ,4H    ,4H    ,4H    ,4H    ,4H    ,4H    /
      DATA TITLE3  /4H    ,4H    ,4H    ,4H    ,4H    ,4H    ,4H    ,
     C4H    ,4H    ,4H    ,4H    ,4H    ,4H    ,4H    ,4H    /
      END
      SUBROUTINE PLTIF(XVMIN,XVMAX,YVMIN,YVMAX)
      COMMON/DATA1/XMARG,XWMAX,XINF,YINF,XLINF,YWMAX,XYMAR,ASPR
      COMMON/DATA2/IPLX,IPLZ,THETA,PHI,DELTA,XSC,YSC,ZSC,SIZE
      COMMON/DATA3/IPLP,INFORM,ITITLE,IPERIM,IAXIS,IMASK
      COMMON/DATA4/XMIN,XMAX,ZMIN,ZMAX,YMIN,YMAX
      COMMON/DATA5/NXPL,NZPL,IPLPX,IPLPZ,XSCAL0,YSCAL0,ZSCAL0
      COMMON/DATA7/NHEDX,HEDX(5),NHEDZ,HEDZ(5),NHEDY,HEDY(5)
      DATA ANO/2HNO/,AYE/2HYE/
         YSP=0.03*YVMAX
```

```
      DX1=.10*XVMAX
      DX2=.125*XVMAX
      DX3=.07*XVMAX
       X0=XVMIN
       Y0=YVMIN
      XOR=XINF
      YOR=YINF
    X = XOR
    Y = YOR
    CALL LETTR(X,Y,7HX MIN =,0.0,7)
    Y=Y-YSP
    CALL LETTR(X,Y,7HX MAX =,0.0,7)
    Y=Y-YSP
    CALL LETTR(X,Y,7HZ MIN =,0.0,7)
    Y=Y-YSP
    CALL LETTR(X,Y,7HZ MAX =,0.0,7)
    Y=Y-YSP
    CALL LETTR(X,Y,7HY MIN =,0.0,7)
    Y=Y-YSP
    CALL LETTR(X,Y,7HY MAX =,0.0,7)
    X=X+DX1
    Y=YOR
    CALL NUMBR(X,Y,XMIN,0.0,2)
    Y=Y-YSP
    CALL NUMBR(X,Y,XMAX,0.0,2)
    Y=Y-YSP
    CALL NUMBR(X,Y,ZMIN,0.0,2)
    Y=Y-YSP
    CALL NUMBR(X,Y,ZMAX,0.0,2)
    Y=Y-YSP
    CALL NUMBR(X,Y,YMIN,0.0,2)
    Y=Y-YSP
    CALL NUMBR(X,Y,YMAX,0.0,2)
      X1=.33*XVMAX
      Y1=YVMAX
    CALL BOXIN(X0,Y0,X1,Y1,0.0,0.0)
     XOR=X1+XMARG
       X=XOR
       Y=YOR
    CALL LETTR(X,Y,7HTHETA =,0.0,7)
       Y=Y-YSP
    CALL LETTR(X,Y,7H  PHI =,0.0,7)
       Y=Y-YSP
    CALL LETTR(X,Y,7H SIZE =,0.0,7)
       Y=Y-YSP
    CALL LETTR(X,Y,7H  XSC =,0.0,7)
       Y=Y-YSP
    CALL LETTR(X,Y,7H  YSC =,0.0,7)
       Y=Y-YSP
    CALL LETTR(X,Y,7H  ZSC =,0.0,7)
       X=X+DX2
       Y=YOR
    CALL NUMBR(X,Y,THETA,0.0,1)
       Y=Y-YSP
```

```
      CALL NUMBR(X,Y,PHI,0.0,1)
         Y=Y-YSP
      CALL NUMBR(X,Y,SIZE,0.0,1)
         Y=Y-YSP
       XSCO=XSC*SIZE
       YSCO=YSC*SIZE
       ZSCO=ZSC*SIZE
      CALL NUMBR(X,Y,XSCO,0.0,1)
         Y=Y-YSP
      CALL NUMBR(X,Y,YSCO,0.0,1)
         Y=Y-YSP
      CALL NUMBR(X,Y,ZSCO,0.0,1)
        X2=X1+.25*XVMAX
        Y1=YVMAX
      CALL BOXIN(X1,Y0,X2,Y1,0.0,0.0)
       XOR=X2+XMARG
       YOR=YOR-.01*YVMAX
       YSP=0.04*YVMAX
   50      CALL ZALPH(38,38H DO YOU WANT NEW TITLES FOR THE AXES  )
           CALL ZKYBD(1,4,NANSW,ANSW)
           IF(ANSW.EQ.ANO) GO TO 100
           IF(ANSW.NE.AYE) GO TO 50
           CALL ZALPH(35,35H TITLE FOR THE Y-AXIS (20 Chars.)  )
           CALL ZKYBD(1,20,NHEDY,HEDY)
           CALL ZALPH(35,35H TITLE FOR THE X-AXIS (20 Chars.)  )
           CALL ZKYBD(1,20,NHEDX,HEDX)
           CALL ZALPH(35,35H TITLE FOR THE Z-AXIS (20 Chars.)  )
           CALL ZKYBD(1,20,NHEDZ,HEDZ)
   10  FORMAT(A2)
   20  FORMAT(20A4)
  100  X = XOR
       Y = YOR
       CALL LETTR(X,Y,4HY = ,0.0,4)
       Y=Y-YSP
       CALL LETTR( X,Y,4HX = ,0.0,4 )
       Y=Y-YSP
       CALL LETTR(X,Y,4HZ = ,0.0,4)
       Y=Y-1.5*YSP
       CALL LETTR(X,Y,21H PLT3T, Ver. 3D:04:84 ,0.0,21)
       X=X+DX3
       Y=YOR
       CALL LETTR(X,Y,HEDY(1),0.0,NHEDY)
       Y=Y-YSP
       CALL LETTR( X,Y,HEDX(1),0.0,NHEDX)
       Y=Y-YSP
       CALL LETTR( X,Y,HEDZ(1),0.0,NHEDZ)
        X0=X2
        X3=XVMAX
        Y1=YVMAX
       CALL BOXIN(X2,Y0,X3,Y1,0.0,0.0)
       RETURN
       END
       SUBROUTINE DATEP(X0,Y0)
       DIMENSION IDATE(15)
```

```
CALL FTIME(IDATE)
CALL LETTR(X0,Y0,IDATE(6),0.0,20)
RETURN
END
SUBROUTINE REFAX(XC,YC,ZC,XMAX,YMAX,ZMAX)
COMMON/DATA1/XMARG,XWMAX,XINF,YINF,XLINF,YWMAX,XYMAR,ASPR
COMMON/DATA2/IPLX,IPLZ,THETA,PHI,DELTA,XSC,YSC,ZSC,SIZE
DATA RD/57.295779513082/
DATA HT0/.01/,WTH0/0.001/
THETAR=THETA/RD
  PHIR=PHI/RD
    HT=HT0
   WHT=WHT0
SINPH = SIN( PHIR )
COSPH = COS( PHIR )
SINTH = SIN( THETAR )
COSTH = COS( THETAR )
XLMX=XMAX
YLMX=YMAX
ZLMX=ZMAX
SX=XLMX*XSC/ABS(XSC)
SY=YLMX*YSC/ABS(YSC)
SZ=ZLMX*ZSC/ABS(ZSC)
XAXX = SX * COSPH
YAXX = SX * SINPH * SINTH
XAXY = 0.
YAXY = SY * COSTH
XAXZ = SZ * SINPH
YAXZ = -SZ * COSPH * SINTH
X = XC
Y = YC
CALL ZMOVE( X , Y )
X = X + XAXX
Y = Y + YAXX
CALL ZDRAW( X , Y )
X=X+HT
Y=Y-2.0*HT
CALL LETTR(X,Y,1HX,0.0,1)
X = XC
Y = YC
CALL ZMOVE( X , Y )
X = X + XAXY
Y = Y + YAXY
CALL ZDRAW( X , Y  )
X=X+HT
Y=Y-HT
CALL LETTR(X,Y,1HY,0.0,1)
X = XC
Y = YC
CALL ZMOVE( X , Y )
X = X + XAXZ
Y = Y + YAXZ
 CALL ZDRAW( X , Y  )
X=X+HT
```

```
      Y=Y-2.*HT
      CALL LETTR(X,Y,1HZ,0.0,1)
      X = XC
      Y = YC
      CALL ZMOVE( X , Y )
      RETURN
      END
      SUBROUTINE BOXIN(X0,Y0,XMAX,YMAX,XYMAR,ASPR)
       XMAR=ASPR*XYMAR
       YMAR=XYMAR
       X1=X0+XMAR
       Y1=Y0+YMAR
       X2=XMAX-XMAR
       Y2=YMAX-YMAR
      CALL ZMOVE(X1,Y1)
      CALL ZDRAW(X1,Y2)
      CALL ZDRAW(X2,Y2)
      CALL ZDRAW(X2,Y1  )
      CALL ZDRAW(X1,Y1  )
      RETURN
      END
      SUBROUTINE TITLE(XTMIN,XTMAX,YTMIN,YTMAX)
      COMMON/DATA1/XMARG,XWMAX,XINF,YINF,XLINF,YWMAX,XYMAR,ASPR
      COMMON/DATA2/IPLX,IPLZ,THETA,PHI,DELTA,XSC,YSC,ZSC,SIZE
      COMMON/DATA4/XMIN,XMAX,ZMIN,ZMAX,YMIN,YMAX
      COMMON/DATA8/NTITL1,TITLE1(2),NTITL2,TITLE2(15),NTITL3,TITLE3(15)
      DATA ANO/2HNO/AYE/2HYE/
         YSP=0.5*YTMAX
 10      CALL ZALPH(31,31H DO YOU WANT NEW FIGURE TITLE  )
         CALL ZKYBD(1,4,NANSW,ANSW)
         IF(ANSW.EQ.ANO) GO TO 100
         IF(ANSW.NE.AYE) GO TO 10
         CALL ZALPH(31,31HTYPE THE 1ST LINE (60 Chars.)  )
         CALL ZKYBD(1,60 ,NTITL2,TITLE2)
         CALL ZALPH(31,31HTYPE THE 2ND LINE (60 Chars.)  )
         CALL ZKYBD(1,60,NTITL3,TITLE3)
100   X0=XTMIN
      Y0=YTMIN
       XOR=X0+XMARG
       YOR=YTMAX-XMARG
        X=XOR
        Y=YOR
      CALL LETTR(X,Y,TITLE2(1),0.0,NTITL2)
      Y=Y-YSP
      CALL LETTR(X,Y,TITLE3(1),0.0,NTITL3)
      Y=Y-YSP
      CALL BOXIN(X0,Y0,XTMAX,YTMAX,0.0,0.0)
      RETURN
      END
      SUBROUTINE DEVON
      DIMENSION SUBR(3)
      DATA SUBR/4H    ,4H    ,4H    /
      LU=1
      CALL ZBEGN
```

```
      CALL ZAINT(1,IERROR)
      IF(IERROR.EQ.0) GO TO 100
      CALL ERRORS(1,IERROR,6HZAINT )
C     GO TO 999
  100 CALL ZKINT(1,IERROR)
      IF(IERROR.EQ.0) GO TO 200
      CALL ERRORS(1,IERROR,6HZKINT )
C     GO TO 999
  200 CALL ZBINT(1,IERROR)
      IF(IERROR.EQ.0) GO TO 400
      CALL ERRORS(1,IERROR,6HZBINT )
C     GO TO 999
  400 ICOTR=0
      CALL ZDINT(LU,ICOTR,IERROR)
      IF(IERROR.EQ.0) GO TO 600
      CALL ERRORS(1,IERROR,6HZDINT )
C     GO TO 999
  600 CONTINUE
  999 RETURN
      END
      SUBROUTINE ERRORS(LU,IERROR,SUBR)
      DIMENSION SUBR(3)
      CALL ZMCUR
      WRITE(LU,100) IERROR,SUBR
  100 FORMAT("<<<<< Error ",I5,"  occured in subroutine",3A4,">>>>>")
      RETURN
      END
      SUBROUTINE VIEWP(XWMAX,YWMAX,SIZE,IERROR)
      DIMENSION AR(2),VIEW(4)
      CALL ZIWS(254,0,2,IDUM,AR,IERROR)
        XSIZE=1.0
        YSIZE=AR(2)
      CALL ZASPK(XSIZE,YSIZE)
        WXMIN=0.0
        WXMAX=XWMAX
        WYMIN=0.00
        WYMAX=YWMAX
      CALL ZWIND(WXMIN,WXMAX,WYMIN,WYMAX)
      CALL ZIWS(451,0,4,IDUM,VIEW,IERROR)
        VMINX=.05*VIEW(2)/SIZE
        VMAXX=.95*VIEW(2)/SIZE
        VMINY=.05*VIEW(4)/SIZE
        VMAXY=.95*VIEW(4)/SIZE
      CALL ZVIEW(VMINX,VMAXX,VMINY,VMAXY)
        XCSIZ=0.015*(WXMAX-WXMIN)/SIZE
        YCSIZ=0.025*(WYMAX-WYMIN)/SIZE
      CALL ZCSIZ(XCSIZ,YCSIZ)
      RETURN
      END
      SUBROUTINE VPMAX(VIEW)
      DIMENSION AR(2),VIEW(4)
      CALL ZIWS(451,0,4,IDUM,VIEW,IERROR)
      CALL ZIWS(254,0,2,IDUM,AR,IERROR)
        XNEW=1.0
```

```
      YNEW=1.0
      IF(AR(2).LE.1.0) YNEW=AR(2)
      IF(AR(2).GT.1.0) XNEW=1./AR(2)
      CALL ZVIEW(0.0,XNEW,0.0,YNEW)
      RETURN
      END
      SUBROUTINE DEVOF
      CALL ZCOLR(0)
      CALL ZAEND
      CALL ZBEND
      CALL ZKEND
      CALL ZDEND
      CALL ZLEND
      CALL ZEND
      RETURN
      END
      SUBROUTINE LETTR (X,Y,IAR,ANG,NC)
      DIMENSION IAR(1),VIEW(4)
      CALL ZMOVE(X,Y)
      CALL VPMAX(VIEW)
      CALL ZTEXT(NC,IAR)
      CALL ZVIEW(VIEW(1),VIEW(2),VIEW(3),VIEW(4))
      RETURN
      END
      SUBROUTINE NUMBR(X,Y,FPN,THETA,IFMT)
      DIMENSION JFMT(5),ISYM(3)
      GO TO (10,20,30) IFMT
   10 ENCODE(15,15,ISYM) FPN
      GO TO 100
   20 ENCODE(15,25,ISYM) FPN
      GO TO 100
   30 ENCODE(10,35,ISYM) FPN
  100 CALL LETTR(X,Y,ISYM,THETA,15)
   15 FORMAT(F5.1)
   25 FORMAT(1PE12.4)
   35 FORMAT(F5.0)
      RETURN
      END
```

```
FTN7X.O
$FILES 0,1
$EMA/BLK1/
      PROGRAM PLT3P(S), G.A.KERAMIDAS 3P:04:25:84
      COMMON/BLK1/MASK(9000),XPL(151),ZPL(151),YPL(151,151)
      COMMON/DATA1/XMARG,XWMAX,XINF,YINF,XLINF,YWMAX,XYMAR,ASPR
      COMMON/DATA2/IPLX,IPLZ,THETA,PHI,DELTA,XSC,YSC,ZSC,SIZE
      COMMON/DATA3/IPLP,INFORM,ITITLE,IPERIM,IAXIS,IMASK
      COMMON/DATA4/XMIN,XMAX,ZMIN,ZMAX,YMIN,YMAX
      COMMON/DATA5/NXPL,NZPL,IPLPX,IPLPZ,XSCALO,YSCALO,ZSCALO
      COMMON/DATA6/INSTN,IFAST,IPLOT,IPMENU,IPEN(2)
      COMMON/DATA7/NHEDX,HEDX(5),NHEDZ,HEDZ(5),NHEDY,HEDY(5)
      COMMON/DATA8/NTITL1,TITLE1(2),NTITL2,TITLE2(15),NTITL3,TITLE3(15)
      DIMENSION VIEW(4)
      Y0=0.0
      X0=0.0
      CALL DEVON
      CALL VIEWP(XWMAX,YWMAX,SIZE,IERROR)
      CALL ZIWS(451,0,4,IDUM,VIEW,IERROR)
      IF(IERROR.EQ.0) GO TO 305
      CALL ERRORS(1,IERROR,10HZIWS. 451 )
      GO TO 999
  305 IF(INFORM.LE.0) GO TO 310
        Y1=0.8*YWMAX
       YPMAX=.2*VIEW(4)
      CALL PLTIF(X0,XWMAX,Y1,YWMAX)
  310 IF(ITITLE.LE.0) GO TO 315
       Y2=0.08*YWMAX
      CALL TITLE(X0,XWMAX,Y0,Y2 )
       Y0=.08*VIEW(4)
  315  CONTINUE
      CALL ZIWS(451,0,4,IDUM,VIEW,IERROR)
      IF(IERROR.EQ.0) GO TO 400
      CALL ERRORS(1,IERROR,10HZIWS. 451 )
      GO TO 999
  400    XV0=VIEW(1)
         YV0=VIEW(3)+Y0
       XVMAX=VIEW(2)
       YVMAX=VIEW(4)-YPMAX
  410 CALL ZVIEW(XV0,XVMAX,YV0,YVMAX)
      CALL ADISP
      CALL ZALPH (40,40H  ******PLOTTING.....!!!!********       )
      CALL ZCOLR(1)
      CALL BOXIN(.0,.0,XWMAX,YWMAX,0.0,0.0)
      CALL ZCOLR(IPEN(2))
      CALL PLOTS(MASK,XPL,ZPL,YPL,NXPL,NZPL,1,NXPL)
      CALL ZMCUR
  500 CONTINUE
  999 CONTINUE
      CALL DEVOF
      CALL SEGRT (Z991)
      END
      SUBROUTINE ADISP
      INTEGER ILIST(7), STRING(2), IERR
```

```
      DATA STRING /15550B,15512B/
C
      CALL ZALPH (4,STRING)
C
      RETURN
      END
      SUBROUTINE PLOTS(MASK,X,Z,Y,NX,NZ,IDIM,IDIMEN)
      LOGICAL LPLOTX,LPLOTZ
      DIMENSION XX(151),ZZ(151),YY(151),VERTEX(16)
      EMA MASK(9000),X(151),Z(151),Y(151,151)
      COMMON/DATA1/XMARG,XWMAX,XINF,YINF,XLINF,YWMAX,XYMAR,ASPR
      COMMON/DATA2/IPLX,IPLZ,THETA,PHI,DELTA,XSC,YSC,ZSC,SIZE
      COMMON/DATA3/IPLP,INFORM,ITITLE,IPERIM,IAXIS,IMASK
      COMMON/DATA5/NXPL,NZPL,IPLPX,IPLPZ,XSCALO,YSCALO,ZSCALO
      COMMON/DATA4/XMIN,XMAX,ZMIN,ZMAX,YMIN,YMAX
      DATA SW/-1./
      PID180=.01745329252
      PIR=3.14159
      COSPH = COS( PID180 * PHI )
      COSTH = COS( PID180 * THETA )
      SINPH = SIN( PID180 * PHI )
      SINTH = SIN( PID180 * THETA )
      TMP1 = XSC * COSPH
      TMP2 = ZSC * SINPH
      XMX  = AMAX1( TMP1 , TMP2 , TMP1+TMP2 , 0. )
      XMN  = AMIN1( TMP1 , TMP2 , TMP1+TMP2 , 0. )
      TMP1 = XSC * SINPH *SINTH
      TMP2 = -ZSC *COSPH *SINTH
      YMX  = AMAX1( TMP1 , TMP2 , TMP1+TMP2 , 0. )
      YMN  = AMIN1( TMP1 , TMP2 , TMP1+TMP2 , 0. )
      TMP1 = YSC * COSTH
      YMX  = AMAX1( YMX , YMX + TMP1 )
      YMN  = AMIN1( YMN , YMN + TMP1 )
      XL = XMX - XMN
      YL = YMX - YMN
      IF( XL .GT. XWMAX  .OR.  YL .GT. YWMAX ) GO TO 998
      XOR = XMARG - XMN
      XMX = XMARG + XL + XMARG
      SW=-SW
      YOR = ABS(YMN)+(YWMAX-YL)/2.+DELTA
      LPLOTX = -SINPH * COSTH .GT. 0.
      LPLOTZ =  COSPH * COSTH .GT. 0.
      XSCALE = XSC/(XMAX-XMIN)
      YSCALE = YSC/(YMAX-YMIN)
      ZSCALE = ZSC/(ZMAX-ZMIN)
      CALL ZMOVE( XOR , 0.0)
      KPLX = MAX0( IPLX , 1 )
      NEXTX = 1
      DO 450 NLINE = 1,NX
      IF( NLINE .NE. NEXTX ) GO TO 450
      NEXTX = MIN0( NLINE + KPLX , NX )
      NN = NLINE
      IF( LPLOTX ) NN = NX + 1 - NLINE
      DO 400 J=1 , NZ
```

```
      XX(J) = X(NN)-XMIN
      YY(J) = Y(NN,J)-YMIN
      ZZ(J) = Z(J)-ZMIN
  400 CONTINUE
      NN = NLINE
      CALL PLOTF( 1111,XX,YY,ZZ,XSCALE,YSCALE,ZSCALE,NN,NZ,-PHI,-THETA,
     *          XOR,YOR,XMX,MASK,VERTEX)
  450 CONTINUE
      IF(IPERIM.GE.1)
     *CALL PERIM(X,Y,Z,XX,YY,ZZ,XSCALE,YSCALE,ZSCALE,XMIN,YMIN,ZMIN,
     1NX,NZ,PHI,THETA,XOR,YOR,XMX,MASK,NN,1)
      KPLZ = MAX0( IPLZ , 1 )
      NEXTZ = 1
      DO 550 NLINE = 1 , NZ
      IF( NLINE .NE. NEXTZ ) GO TO 550
      NEXTZ = MIN0( NLINE + KPLZ , NZ )
      NN = NLINE
      IF( LPLOTZ ) NN = NZ + 1 - NLINE
      DO 500  I = 1 , NX
      XX(I) = X(I)-XMIN
      YY(I) = Y(I,NN)-YMIN
      ZZ(I) = Z(NN)-ZMIN
  500 CONTINUE
      NN = NLINE
      CALL PLOTF( 0111,XX,YY,ZZ,XSCALE,YSCALE,ZSCALE,NN,NX,-PHI,-THETA,
     *          XOR,YOR,XMX,MASK,0.0)
  550 CONTINUE
      IF(IPERIM.GE.1)
     *CALL PERIM(X,Y,Z,XX,YY,ZZ,XSCALE,YSCALE,ZSCALE,XMIN,YMIN,ZMIN,
     1NX,NZ,PHI,THETA,XOR,YOR,XMX,MASK,NN,2)
         XC0=XWMAX-4.*XMARG
         YC0=.85*YWMAX
         ZC0=XC0
        XLMX=2.5*XMARG
        YLMX=3.0*XMARG
        ZLMX=2.5*XMARG
       CALL ZCOLR(1)
      IF(IAXIS.GE.1)   CALL REFAX(XC0,YC0,ZC0,XLMX,YLMX,ZLMX)
       XP0=XMARG
       YP0=.95*YWMAX
      CALL DATEP(XP0,YP0)
      GO TO 999
  998 CONTINUE
      CALL LETTR(0.2,0.25,.02,26HFIGURE SIZE LIMIT EXCEEDED,0.,26 )
  999  CONTINUE
      RETURN
      END
      SUBROUTINE PLOTF(IVXYZ,XDATA,YDATA,ZDATA,XSCALE,
     *YSCALE,ZSCALE,NLINE,NPNTS,PHI,THETA,XREF,
     *YREF,XLENTH,MASK,VERTEX)
       INTEGER HIGH, OLDHI, OLDLOW
       COMMON/DATA2/IPLX0,IPLZ0,THETA0,PHI0,DELTA0,XSC0,YSC0,ZSC0,SIZE
       COMMON/DATA3/IPLP,INFORM,ITITLE,IPERIM,IAXIS,IMASK
       COMMON/DATA5/NXPL,NZPL,IPLPX,IPLPZ,XSCAL0,YSCAL0,ZSCAL0
```

```
      DIMENSION XDATA(1), YDATA(1), ZDATA(1),VERTEX(16)
      EMA MASK(9000)
      DATA INIT, JVXYZ, SPHI, STHETA/-1,-1,-1.0E30,-1.0E30/
      IF (NLINE.EQ.0) GO TO 550
      IF (NLINE.NE.1) GO TO 20
      PIPI=100.*SQRT(SIZE)
      NYPI=PIPI*20
      LIMITX=(XLENTH*PIPI)+0.5
      I=LIMITX+LIMITX
      DO 10 K=1,I
      MASK(K)=INIT
10    CONTINUE
      INIT=-1
      INCI=-1
      I=0
 20   IF(JVXYZ.EQ.IVXYZ)GO TO 70
      JVXYZ=IVXYZ
      INDZ=1
      INDY=1
      INDX=1
      INDV=1
      IF(JVXYZ.LT.1000)GO TO 30
      INDV=2
      JVXYZ=JVXYZ-1000
 30   IF(JVXYZ.LT.100)GO TO 40
      INDX=2
      JVXYZ=JVXYZ-100
 40   IF(JVXYZ.LT.10)GO TO 50
      INDY=2
      JVXYZ=JVXYZ-10
 50   IF(JVXYZ.LT.1)GO TO 60
      INDZ=2
 60   JVXYZ=IVXYZ
 70   IF (PHI.EQ.SPHI.AND.THETA.EQ.STHETA) GO TO 80
      SPHI=SIN(.0174532925*PHI)
      CPHI=COS(.0174532925*PHI)
      STHETA=SIN(.0174532925*THETA)
      CTHETA=COS(.0174532925*THETA)
      A11=CPHI
      A13=-SPHI
      A21=STHETA*SPHI
      A22=CTHETA
      A23=STHETA*CPHI
      SPHI=PHI
      STHETA=THETA
 80   INCI=-INCI
      IF(I.NE.0)I=NPNTS+1
      DO 530 K=1,NPNTS
        I=I+INCI
        GO TO (90,100),INDX
 90   X=XDATA(1)+(I-1)*XSCALE
        GO TO 110
100     X=XDATA(I)*XSCALE
110     GO TO (120,130),INDY
```

```
120 Y=YDATA(1)+(I-1)*YSCALE
      GO TO 140
130   Y=YDATA(I)*YSCALE
140   GO TO (150,160),INDZ
150 Z=ZDATA(1)+(NLINE-1)*ZSCALE
      GO TO 170
160   Z=ZDATA(I)*ZSCALE
170   XXX=A11*X+A13*Z+XREF
      XX=XXX
    IX=(XX*PIPI)+.5
      YYY=A21*X + A23*Z + YREF
      YY=YYY+A22*Y
    IY=(YY*PIPI)+.5
      IF (IX.LE.0)IX=1
      IF(IX.GT.LIMITX)IX=LIMITX
      IF(IY.LT.5)IY=5
      IF(IY.GT.NYPI)IY=NYPI
      IF (K.NE.1)GO TO 250
      LOW=IX+IX
      HIGH =LOW-1
      MLOW=MASK(LOW)
      MHIGH=MASK(HIGH)
      IF(MHIGH-IY)200,210,180
180   IF(MLOW-IY)190,230,220
190   LOCOLD=0
      GOTO 240
200   MASK(HIGH)=IY
      IF(MLOW.EQ.-1)MASK(LOW)=IY
210   LOCOLD=+1
      GO TO 240
220   MASK(LOW)=IY
230   LOCOLD=-1
240   IF(IMASK) 241,241,242
241   CALL ZMOVE(XX,YY)
      GO TO 243
242  IF(LOCOLD.NE.0) CALL ZMOVE(FLOAT(IX)/PIPI,FLOAT(IY)/PIPI)
243   CONTINUE
      JX=IX
      JY=IY
      IYREF=IY
      IF(INDV.EQ.1)GO TO 530
      INDEX=INCI+6
      VERTEX(INDEX)=XX
      VERTEX(INDEX+1)=YY
      VERTEX(INDEX+8)=XXX
      VERTEX(INDEX+9)=YYY
      IF(NLINE.NE.1) GO TO 530
      VERTEX(1)=XX
      VERTEX(2)=YY
      VERTEX(9)=XXX
      VERTEX(10)=YYY
      GO TO 530
250   IF(IMASK.GT.0) GO TO 255
      CALL ZDRAW(XX,YY)
```

```
      GO TO 530
  255 IF(IX.NE.JX)GO TO 260
      JY=IY
      GO TO 280
  260 YINC=FLOAT(IY-JY)/ABS(FLOAT(IX-JX))
      INCX=(IX-JX)/IABS(IX-JX)
      YJ=JY
  270 JX=JX+INCX
      YJ=YJ+YINC
     JY=YJ+.5
      LOW=JX+JX
      HIGH=LOW-1
      MLOW=MASK(LOW)
      MHIGH=MASK(HIGH)
  280 IF(MHIGH-JY)300,300,290
  290 IF(MLOW-JY)310,320,320
  300 LOC=+1
      IF(LOCOLD)360,370,430
  310 LOC=0
      IF(LOCOLD)340,350,330
  320 LOC=-1
      IF(LOCOLD)510,450,440
  330 IF(MHIGH.LE.IYREF)CALL ZDRAW(FLOAT(JX)/PIPI,FLOAT(MHIGH)/PIPI)
      GO TO 350
  340 IF(MLOW.GE.IYREF)CALL ZDRAW(FLOAT(JX)/PIPI,FLOAT(MLOW)/PIPI)
  350 CONTINUE
      GO TO 520
  360 IF(MLOW-IYREF)370,380,380
  370 IF(MHIGH-IYREF)400,390,390
  380 CALL ZDRAW(FLOAT(JX)/PIPI,FLOAT(MLOW)/PIPI)
  390 CALL ZMOVE(FLOAT(JX)/PIPI,FLOAT(MHIGH)/PIPI)
      GO TO 430
  400 IF(MHIGH.EQ.-1)GO TO 430
      OLDHI=HIGH-2*INCX
      IF(MASK(OLDHI)-JY)420,420,410
  410 CALL ZMOVE(FLOAT(JX)/PIPI,FLOAT(JY)/PIPI)
      GO TO 430
  420 IP=JX-INCX
     ID=MASK(OLDHI)
     CALL ZMOVE(FLOAT(IP)/PIPI,FLOAT(ID)/PIPI)
  430 MASK(HIGH)=JY
      IF(MLOW.EQ.-1)MASK(LOW)=JY
     CALL ZDRAW(FLOAT(JX)/PIPI,FLOAT(JY)/PIPI)
      GO TO 520
  440 IF(MHIGH-IYREF)460,460,450
  450 IF(MLOW-IYREF)470,470,480
  460 CALL ZDRAW(FLOAT(JX)/PIPI,FLOAT(MHIGH)/PIPI)
  470 CALL ZMOVE(FLOAT(JX)/PIPI,FLOAT(MLOW)/PIPI)
      GO TO 510
  480 OLDLOW=LOW-2*INCX
      IF(MASK(OLDLOW)-JY)490,500,500
  490 CALL ZMOVE(FLOAT(JX)/PIPI,FLOAT(JY)/PIPI)
      GO TO 510
  500 IP=JX-INCX
```

```
      ID=MASK(OLDLOW)
      CALL ZMOVE(FLOAT(IP)/PIPI,FLOAT(ID)/PIPI)
  510   MASK(LOW)=JY
      CALL ZDRAW(FLOAT(JX)/PIPI,FLOAT(JY)/PIPI)
  520   IYREF=JY
        LOCOLD=LOC
        IF(JX.NE.IX) GO TO 270
  530 CONTINUE
      IF(IMASK) 531,531,532
  531 CALL ZMOVE(XX,YY)
      GO TO 535
  532 CALL ZMOVE(FLOAT(JX)/PIPI,FLOAT(JY)/PIPI)
  535 IF(INDV.EQ.1)GO TO 540
      INDEX=-INCI+6
      VERTEX(INDEX)=XX
      VERTEX(INDEX+1)=YY
      VERTEX(INDEX+8)=XXX
      VERTEX(INDEX+9)=YYY
      IF (NLINE.NE.1)GO TO 540
      VERTEX(3)=XX
      VERTEX(4)=YY
      VERTEX(11)=XXX
      VERTEX(12)=YYY
  540 I=I-1
      RETURN
  550 INIT=0
      RETURN
      END
      SUBROUTINE PERIM(X,Y,Z,XX,YY,ZZ,XSCALE,YSCALE,ZSCALE,XMIN,YMIN,
     *                 ZMIN,NX,NZ,PHI,THETA,XOR,YOR,XMX,MASK,NN,IDIM)
      COMMON/DATA3/IPLP,INFORM,ITITLE,IPERIM,IAXIS,IMASK
      COMMON/DATA5/NXPL,NZPL,IPLPX,IPLPZ,XSCAL0,YSCAL0,ZSCAL0
      DIMENSION XX(1),YY(1),ZZ(1)
      COMMON/DATA6/INSTN,IFAST,IPLOT,IPMENU,IPEN(2)
      DIMENSION XV(2),YV(2),ZV(2)
      EMA MASK(9000),X(151),Z(151),Y(151,151)
      IDIREC=-1
      IF(IPLPX.GT.0) GO TO 2
      IPLPX=NX-1
      IPLPZ=NZ-1
    2 GO TO (4,11),IDIM
    4 DO 5 J=1,NZ
      XX(J)=X(1)-XMIN
      YY(J)=0.0
      ZZ(J)=Z(J)-ZMIN
    5 CONTINUE
      DO 10 J=1,NZ,IPLPZ
      IDIREC=-IDIREC
      XV(1)=XX(J)
      XV(2)=XV(1)
      YV(1)=0.0
      YV(2)=0.0
      IF(IDIREC.EQ.-1) YV(1)=Y(1,J)-YMIN
      IF(IDIREC.EQ.+1) YV(2)=Y(1,J)-YMIN
```

```
      ZV(1)=ZZ(J)
      ZV(2)=ZV(1)
   10 CALL PLOTF(0111,XV,YV,ZV,XSCALE,YSCALE,ZSCALE,NN,2,-PHI,
     * -THETA,XOR,YOR,XMX,MASK,0.0)
      CALL PLOTF(0111,XX,YY,ZZ,XSCALE,YSCALE,ZSCALE,NN,NZ,-PHI,
     * -THETA,XOR,YOR,XMX,MASK,0.0)
      GO TO 21
   11 DO 15 I=1,NX
      XX(I)=X(I)-XMIN
      YY(I)=0.0
      ZZ(I)=Z(NZ)-ZMIN
   15 CONTINUE
      DO 20 I=1,NX,IPLPX
      IDIREC=-IDIREC
      XV(1)=XX(I)
      XV(2)=XV(1)
      YV(1)=0.0
      YV(2)=0.0
      IF(IDIREC.EQ.-1) YV(1)=Y(I,NZ)-YMIN
      IF(IDIREC.EQ.+1) YV(2)=Y(I,NZ)-YMIN
      ZV(1)=ZZ(I)
      ZV(2)=ZV(1)
   20 CALL PLOTF(0111,XV,YV,ZV,XSCALE,YSCALE,ZSCALE,NN,2,-PHI,
     * -THETA,XOR,YOR,XMX,MASK,0.0)
      CALL PLOTF(0111,XX,YY,ZZ,XSCALE,YSCALE,ZSCALE,NN,NX,-PHI,
     * -THETA,XOR,YOR,XMX,MASK,0.0)
      GO TO 31
   21 DO 25 J=1,NZ
      JJ=NZ+1-J
      XX(J)=X(NX)-XMIN
      YY(J)=0.0
      ZZ(J)=Z(JJ)-ZMIN
   25 CONTINUE
      DO 30 J=1,NZ,IPLPZ
      IDIREC=-IDIREC
      JJ=NZ+1-J
      XV(1)=XX(J)
      XV(2)=XV(1)
      YV(1)=0.0
      YV(2)=0.0
      IF(IDIREC.EQ.-1) YV(1)=Y(NX,JJ)-YMIN
      IF(IDIREC.EQ.+1) YV(2)=Y(NX,JJ)-YMIN
      ZV(1)=ZZ(J)
      ZV(2)=ZV(1)
   30 CALL PLOTF(0111,XV,YV,ZV,XSCALE,YSCALE,ZSCALE,NN,2,-PHI,
     * -THETA,XOR,YOR,XMX,MASK,0.0)
      CALL PLOTF(0111,XX,YY,ZZ,XSCALE,YSCALE,ZSCALE,NN,NZ,-PHI,
     * -THETA,XOR,YOR,XMX,MASK,0.0)
      GO TO 41
   31 DO 35 I=1,NX
      II=NX+1-I
      XX(I)=X(II)-XMIN
      YY(I)=0.0
      ZZ(I)=Z(1)-ZMIN
```

```
 35  CONTINUE
     DO 40 I=1,NX,IPLPX
     II=NX+1-I
     IDIREC=-IDIREC
     XV(1)=XX(I)
     XV(2)=XV(1)
     YV(1)=0.0
     YV(2)=YV(1)
     IF(IDIREC.EQ.-1) YV(1)=Y(II,1)-YMIN
     IF(IDIREC.EQ.+1) YV(2)=Y(II,1)-YMIN
     ZV(1)=ZZ(I)
     ZV(2)=ZV(1)
 40  CALL PLOTF(0111,XV,YV,ZV,XSCALE,YSCALE,ZSCALE,NN,2,-PHI,
    * -THETA,XOR,YOR,XMX,MASK,0.0)
     CALL PLOTF(0111,XX,YY,ZZ,XSCALE,YSCALE,ZSCALE,NN,NX,-PHI,
    * -THETA,XOR,YOR,XMX,MASK,0.0)
 41  CONTINUE
100  RETURN
     END
     SUBROUTINE PLTIF(XVMIN,XVMAX,YVMIN,YVMAX)
     COMMON/DATA1/XMARG,XWMAX,XINF,YINF,XLINF,YWMAX,XYMAR,ASPR
     COMMON/DATA2/IPLX,IPLZ,THETA,PHI,DELTA,XSC,YSC,ZSC,SIZE
     COMMON/DATA3/IPLP,INFORM,ITITLE,IPERIM,IAXIS,IMASK
     COMMON/DATA4/XMIN,XMAX,ZMIN,ZMAX,YMIN,YMAX
     COMMON/DATA5/NXPL,NZPL,IPLPX,IPLPZ,XSCAL0,YSCAL0,ZSCAL0
     COMMON/DATA6/INSTN,IFAST,IPLOT,IPMENU,IPEN(2)
     COMMON/DATA7/NHEDX,HEDX(5),NHEDZ,HEDZ(5),NHEDY,HEDY(5)
     DATA AND/2HND/AYE/2HYE/
        YSP=0.03*YVMAX
        DX1=.10*XVMAX
        DX2=.125*XVMAX
        DX3=.07*XVMAX
         X0=XVMIN
         Y0=YVMIN
        XOR=XINF
        YOR=YINF
      X = XOR
      Y = YOR
      CALL LETTR(X,Y,7HX MIN =,0.0,7)
      Y=Y-YSP
      CALL LETTR(X,Y,7HX MAX =,0.0,7)
      Y=Y-YSP
      CALL LETTR(X,Y,7HZ MIN =,0.0,7)
      Y=Y-YSP
      CALL LETTR(X,Y,7HZ MAX =,0.0,7)
      Y=Y-YSP
      CALL LETTR(X,Y,7HY MIN =,0.0,7)
      Y=Y-YSP
      CALL LETTR(X,Y,7HY MAX =,0.0,7)
      X=X+DX1
      Y=YOR
      CALL NUMBR(X,Y,XMIN,0.0,2)
      Y=Y-YSP
      CALL NUMBR(X,Y,XMAX,0.0,2)
```

```
Y=Y-YSP
CALL NUMBR(X,Y,ZMIN,0.0,2)
Y=Y-YSP
CALL NUMBR(X,Y,ZMAX,0.0,2)
Y=Y-YSP
CALL NUMBR(X,Y,YMIN,0.0,2)
Y=Y-YSP
CALL NUMBR(X,Y,YMAX,0.0,2)
  X1=.33*XVMAX
  Y1=YVMAX
CALL BOXIN(X0,Y0,X1,Y1,0.0,0.0)
 XOR=X1+XMARG
   X=XOR
   Y=YOR
CALL LETTR(X,Y,7HTHETA =,0.0,7)
   Y=Y-YSP
CALL LETTR(X,Y,7H  PHI =,0.0,7)
   Y=Y-YSP
CALL LETTR(X,Y,7H SIZE =,0.0,7)
   Y=Y-YSP
CALL LETTR(X,Y,7H  XSC =,0.0,7)
   Y=Y-YSP
CALL LETTR(X,Y,7H  YSC =,0.0,7)
   Y=Y-YSP
CALL LETTR(X,Y,7H  ZSC =,0.0,7)
   X=X+DX2
   Y=YOR
CALL NUMBR(X,Y,THETA,0.0,1)
   Y=Y-YSP
CALL NUMBR(X,Y,PHI,0.0,1)
   Y=Y-YSP
CALL NUMBR(X,Y,SIZE,0.0,1)
   Y=Y-YSP
 XSCO=XSC*SIZE
 YSCO=YSC*SIZE
 ZSCO=ZSC*SIZE
CALL NUMBR(X,Y,XSCO,0.0,1)
   Y=Y-YSP
CALL NUMBR(X,Y,YSCO,0.0,1)
   Y=Y-YSP
CALL NUMBR(X,Y,ZSCO,0.0,1)
  X2=X1+.25*XVMAX
  Y1=YVMAX
CALL BOXIN(X1,Y0,X2,Y1,0.0,0.0)
 XOR=X2+XMARG
 YOR=YOR-0.01*YVMAX
 YSP=0.04*YVMAX
X = XOR
Y = YOR
CALL LETTR(X,Y,4HY = ,0.0,4)
Y=Y-YSP
CALL LETTR( X,Y,4HX = ,0.0,4 )
Y=Y-YSP
CALL LETTR(X,Y,4HZ = ,0.0,4)
```

```
Y=Y-1.5*YSP
CALL LETTR(X,Y,21H PLT3P, Ver. 3D:04:84 ,0.0,21)
X=X+DX3
Y=YOR
CALL LETTR(X,Y,HEDY(1),0.0,NHEDY)
Y=Y-YSP
CALL LETTR( X,Y,HEDX(1),0.0,NHEDX)
Y=Y-YSP
CALL LETTR( X,Y,HEDZ(1),0.0,NHEDZ)
 X0=X2
 X3=XVMAX
 Y1=YVMAX
CALL BOXIN(X2,Y0,X3,Y1,0.0,0.0)
RETURN
END
SUBROUTINE DATEP(X0,Y0)
DIMENSION IDATE(15)
CALL FTIME(IDATE)
CALL LETTR(X0,Y0,IDATE(6),0.0,20)
RETURN
END
SUBROUTINE REFAX(XC,YC,ZC,XMAX,YMAX,ZMAX)
COMMON/DATA1/XMARG,XWMAX,XINF,YINF,XLINF,YWMAX,XYMAR,ASPR
COMMON/DATA2/IPLX,IPLZ,THETA,PHI,DELTA,XSC,YSC,ZSC,SIZE
DATA RD/57.295779513082/
DATA HT0/.01/,WTH0/0.001/
THETAR=THETA/RD
  PHIR=PHI/RD
    HT=HT0
   WHT=WHT0
SINPH = SIN( PHIR )
COSPH = COS( PHIR )
SINTH = SIN( THETAR )
COSTH = COS( THETAR )
XLMX=XMAX
YLMX=YMAX
ZLMX=ZMAX
SX=XLMX*XSC/ABS(XSC)
SY=YLMX*YSC/ABS(YSC)
SZ=ZLMX*ZSC/ABS(ZSC)
XAXX = SX * COSPH
YAXX = SX * SINPH * SINTH
XAXY = 0.
YAXY = SY * COSTH
XAXZ = SZ * SINPH
YAXZ = -SZ * COSPH * SINTH
X = XC
Y = YC
CALL ZMOVE( X , Y )
X = X + XAXX
Y = Y + YAXX
CALL ZDRAW( X , Y )
X=X+HT
Y=Y-2.0*HT
```

```
CALL LETTR(X,Y,1HX,0.0,1)
X = XC
Y = YC
CALL ZMOVE( X , Y )
X = X + XAXY
Y = Y + YAXY
CALL ZDRAW( X , Y  )
X=X+HT
Y=Y-HT
CALL LETTR(X,Y,1HY,0.0,1)
X = XC
Y = YC
CALL ZMOVE( X , Y )
X = X + XAXZ
Y = Y + YAXZ
 CALL ZDRAW( X , Y  )
X=X+HT
Y=Y-2.*HT
CALL LETTR(X,Y,1HZ,0.0,1)
X = XC
Y = YC
CALL ZMOVE( X , Y )
RETURN
END
SUBROUTINE BOXIN(X0,Y0,XMAX,YMAX,XYMAR,ASPR)
 XMAR=ASPR*XYMAR
 YMAR=XYMAR
 X1=X0+XMAR
 Y1=Y0+YMAR
 X2=XMAX-XMAR
 Y2=YMAX-YMAR
CALL ZMOVE(X1,Y1)
CALL ZDRAW(X1,Y2)
CALL ZDRAW(X2,Y2)
CALL ZDRAW(X2,Y1  )
CALL ZDRAW(X1,Y1  )
RETURN
END
SUBROUTINE TITLE(XTMIN,XTMAX,YTMIN,YTMAX)
COMMON/DATA1/XMARG,XWMAX,XINF,YINF,XLINF,YWMAX,XYMAR,ASPR
COMMON/DATA2/IPLX,IPLZ,THETA,PHI,DELTA,XSC,YSC,ZSC,SIZE
COMMON/DATA4/XMIN,XMAX,ZMIN,ZMAX,YMIN,YMAX
COMMON/DATA9/NTITL1,TITLE1(2),NTITL2,TITLE2(15),NTITL3,TITLE3(15)
DATA AND/2HND/AYE/2HYE/
   YSP=0.5*YTMAX
X0=XTMIN
Y0=YTMIN
 XOR=X0+XMARG
 YOR=YTMAX-XMARG
  X=XOR
  Y=YOR
CALL LETTR(X,Y,TITLE2(1),0.0,NTITL2)
Y=Y-YSP
CALL LETTR(X,Y,TITLE3(1),0.0,NTITL3)
```

```
      Y=Y-YSP
      CALL BOXIN(X0,Y0,XTMAX,YTMAX,0.0,0.0)
      RETURN
      END
      SUBROUTINE DEVON
      COMMON/DATA6/INSTN,IFAST,IPLOT,IPMENU,IPEN(2)
      DIMENSION SUBR(3)
      DATA SUBR/4H    ,4H    ,4H    /
      LUT=1
      LUP=35
   90 WRITE(1,1095)
 1095 FORMAT("************************************************************"
     C    ,/,"**** SPECIFY THE PEN COLOR FOR THE PLOTTER          ****"
     C    ,/,"**** 1. BLACK, 2. BLUE, 3. RED, 4. GREEN           ****"
     C    ,/,"**** PEN COLOR = ")
      READ(1,*) IPEN(2)
      IF (IPEN(2).LE.0) IPEN(2)=1
      WRITE(1,1100)
 1100 FORMAT("************************************************************"
     C    ,/,"**** SPECIFY THE PEN SPEED FOR THE PLOTTER, (1-35)  ****"
     C    ,/,"**** RECOMMENDED VALUE FOR QUALITY PLOTS : 5        ****"
     C    ,/,"**** PEN SPEED = ")
      READ(1,*) IPEN(1)
      IF (IPEN(1) .LE. 0) IPEN(1)=5
      CALL ZBCON
      CALL ZAINT(LUT,IERROR)
      IF(IERROR.EQ.0) GO TO 100
      CALL ERRORS(LUT,IERROR,6HZAINT )
C     GO TO 999
  100 CALL ZKINT(LUT,IERROR)
      IF(IERROR.EQ.0) GO TO 200
      CALL ERRORS(LUT,IERROR,6HZKINT )
C     GO TO 999
  200 CALL ZBINT(LUT,IERROR)
      IF(IERROR.EQ.0) GO TO 400
      CALL ERRORS(LUT,IERROR,6HZBINT )
C     GO TO 999
  400 ICOTR=0
      CALL ZDINT(LUP,ICOTR,IERROR)
      IF(IERROR.EQ.0) GO TO 600
      CALL ERRORS(LUT,IERROR,6HZDINT )
C     GO TO 999
  600 WRITE(1,101)
  101 FORMAT (" Type ( / ) to continue when ready : ")
      READ (1,*) XN
      CALL ZOESC(2050,2,0,IPEN,RDUM,IERROR)
      IF (IERROR.EQ.0) GO TO 999
      CALL ERRORS(LUT,IERROR,10HZOESC 2050)
  999 RETURN
      END
      SUBROUTINE ERRORS(LU,IERROR,SUBR)
      DIMENSION SUBR(3)
      CALL ZMCUR
      WRITE(LU,100) IERROR,SUBR
```

```
100  FORMAT("<<<<< Error ",I5," occured in subroutine",3A4,">>>>>")
     RETURN
     END
     SUBROUTINE VIEWP(XWMAX,YWMAX,SIZE,IERROR)
     DIMENSION AR(2),VIEW(4)
     CALL ZIWS(254,0,2,IDUM,AR,IERROR)
       XSIZE=1.0
       YSIZE=AR(2)
     CALL ZASPK(XSIZE,YSIZE)
       WXMIN=0.0
       WXMAX=XWMAX
       WYMIN=0.00
       WYMAX=YWMAX
     CALL ZWIND(WXMIN,WXMAX,WYMIN,WYMAX)
     CALL ZIWS(451,0,4,IDUM,VIEW,IERROR)
       VMINX=.05*VIEW(2)/SIZE
       VMAXX=.95*VIEW(2)/SIZE
       VMINY=.05*VIEW(4)/SIZE
       VMAXY=.95*VIEW(4)/SIZE
     CALL ZVIEW(VMINX,VMAXX,VMINY,VMAXY)
       XCSIZ=0.015*(WXMAX-WXMIN)/SIZE
       YCSIZ=0.025*(WYMAX-WYMIN)/SIZE
     CALL ZCSIZ(XCSIZ,YCSIZ)
     RETURN
     END
     SUBROUTINE VPMAX(VIEW)
     DIMENSION AR(2),VIEW(4)
     CALL ZIWS(451,0,4,IDUM,VIEW,IERROR)
     CALL ZIWS(254,0,2,IDUM,AR,IERROR)
       XNEW=1.0
       YNEW=1.0
     IF(AR(2).LE.1.0) YNEW=AR(2)
     IF(AR(2).GT.1.0) XNEW=1./AR(2)
     CALL ZVIEW(0.0,XNEW,0.0,YNEW)
     RETURN
     END
     SUBROUTINE DEVOF
     CALL ZCOLR(0)
     CALL ZAEND
     CALL ZBEND
     CALL ZKEND
     CALL ZDEND
     CALL ZLEND
     CALL ZEND
     RETURN
     END
     SUBROUTINE LETTR (X,Y,IAR,ANG,NC)
     DIMENSION IAR(1),VIEW(4)
     CALL ZMOVE(X,Y)
     CALL VPMAX(VIEW)
     CALL ZTEXT(NC,IAR)
     CALL ZVIEW(VIEW(1),VIEW(2),VIEW(3),VIEW(4))
     RETURN
     END
```

```
      SUBROUTINE NUMBR(X,Y,FPN,THETA,IFMT)
      DIMENSION JFMT(5),ISYM(3)
      GO TO (10,20,30) IFMT
   10 ENCODE(15,15,ISYM) FPN
      GO TO 100
   20 ENCODE(15,25,ISYM) FPN
      GO TO 100
   30 ENCODE(10,35,ISYM) FPN
  100 CALL LETTR(X,Y,ISYM,THETA,15)
   15 FORMAT(F5.1)
   25 FORMAT(1PE12.4)
   35 FORMAT(F5.0)
      RETURN
      END
```

GPO 910-207

END

FILMED

1-85

DTIC